

Sami Tavasti

OPENSTACK-PILVIPALVELUALUSTAN KONFIGUROINTI

OPENSTACK-PILVIPALVELUALUSTAN KONFIGUROINTI

Sami Tavasti
Opinnäytetyö
Kevät 2018
Tietojenkäsittely
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu
Tietojenkäsittely, järjestelmänasiantuntija

Tekijä: Sami Tavasti

Opinnäytetyön nimi: Openstack-pilvipalvelualustan konfigurointi

Työn ohjaaja: Jukka Kaisto

Työn valmistumislukukausi ja -vuosi: Kevät 2018

Sivumäärä: 37 + 21 Liitettä

Viime vuosien aikana pilvipalvelujen käyttö on lisääntynyt. Openstack on yksi suurimmista avoimeen lähdekoodiin perustuvista pilvipalvelualustoista. Rakenteeltaan Openstack-pilvipalvelualusta muodostuu erikseen konfiguroitavista palveluista.

Opinnäytetyön tavoitteena on konfiguroida Openstack-alustan testiympäristö. Openstackin testi-asennus mahdollisti perehtymisen Openstackin toimintaan pilvipalvelualustana. Openstack asennettiin Openstack-säätiön dokumentoinnin perusteella virtuaalikoneilla.

Työn teoriaosassa käydään läpi yleisellä tasolla mitä pilvipalveluilla tarkoitetaan ja miten niitä voidaan jaotella. Käytännön osuudessa Openstack-alustaan perehdytään asentamalla testiympäristön muodostavat palvelut jotka ovat identiteettipalvelu Keystone, levykuvapalvelu Glance, compute-palvelu Nova, verkkopalvelu Neutron, varastointipalvelu Cinder, DNS-palvelu Designate sekä graafinen käyttöliittymä Horizon. Lopuksi ympäristön toimintaa testataan.

Asiasanat: pilvipalvelut Linux virtualisointi

ABSTRACT

Oulu University of Applied Sciences
Degree Programme of Business Information Systems, System Administrator

Author: Sami Tavasti

Title of thesis: Configuration of Openstack Cloud computing platform

Supervisor: Jukka Kaisto

Term and year of completion: Spring 2018

Number of pages: 37 + 21 appendices

During the last few years use of cloud computing has increased. Openstack is one of the largest cloud computing platform based on open source. Structure of Openstack cloud computing platform consist of multiple separately configured services.

Purpose of this bachelor's thesis was to configurate test environment of Openstack platform. Test installation of Openstack helped to make oneself familiar with Openstack as a cloud computing platform. Openstack was installed based on documentation of Openstack Foundation on virtual machines.

Theory part of the thesis tells what Cloud services are and how you can divide them. Practical part of the thesis focuses on installation of services which Openstack consists of. These services are identity service Keystone, image service Glance, compute service Nova, network service Neutron, block storage service Cinder, DNS service Designate and graphical user interface Horizon. Finally, the environment is tested.

Keywords: cloud computing Linux virtualisation

SISÄLLYS

1	JOHDANTO	6
2	PILVIPALVELUT	7
2.1	Julkinen pilvi	7
2.2	Yksityinen pilvi	8
2.3	Hybridipilvi	9
3	PALVELUTYYPIT	10
3.1	Infrastruktuuri palveluna	10
3.2	Alusta palveluna	11
3.3	Sovellus palveluna	11
4	OPENSTACK-PILVIPALVELUALUSTA	12
4.1	Testiympäristö	13
4.2	Alkuvalmistelut	16
4.2.1	Keystone	18
4.2.2	Glance	20
4.2.3	Nova	21
4.2.4	Neutron	23
4.2.5	Cinder	25
4.2.6	Designate	26
4.2.7	Horizon	27
4.3	Ympäristön testaus	28
4.4	Ympäristön hallinta	31
5	POHDINTA	32
	LÄHTEET	33
	LIITTEET	38

1 JOHDANTO

Pilvipalvelujen avulla yritykset voivat valjastaa lisää resursseja käyttöönsä tietoteknisissä ympäristöissä pelkän verkkoyhteyden avulla. Lisäksi pilvipalvelujen vaikutus nykypäivänä muun muassa Internetiä selailemalla on huomattavasti suurempi kuin miltä se ensisilmäyksellä näyttää. Useat verkkosivut hyödyntävät pilvipalveluita sivujen ylläpitämisessä sekä verkkoselaimen kautta käytettäviä työvälineohjelmistoja pidetään yhtenä pilvipalvelutyypeistä. Openstack, jonka testiasennus tässä opinnäytetyössä käydään läpi, on yksi suurimmista avoimen lähdekoodin ohjelmistoista pilvipalvelukokonaisuuksien hyödyntämiseen (Openstack Foundation 2018a, viitattu 29.3.2018).

Pilvipalvelut voidaan jakaa erilaisiin pilvi- sekä palvelutyyppeihin. Pilvet voivat olla joko erillisen palvelun toimittajan ylläpitämiä kokonaisuuksia tai yritysten itse pystyttämiä. Palvelutyyppeiden avulla pilvipalveluja käyttävä yritys voi valita, kuinka suuresta kokonaisuudesta yritys itse haluaa vastata.

Saksalaisen ohjelmistoyritys SUSEn toimeksiannosta vuonna 2015 teetetyn pilvipalveluita koskevan kansainvälisen kyselyn mukaan 81 % suurista yrityksistä käyttää tai aikoo käyttää Openstack-pohjaisia ratkaisuja omissa ympäristöissään (Suse 2016, viitattu 17.5.2018). Luku on todennäköisesti tällä hetkellä jopa suurempi. Openstack voidaan luokitella palvelutyypiltään infrastruktuuri palveluna -ratkaisuksi. Openstack-alustan avulla on mahdollista muun muassa ajaa virtuaalikoneita ja hallita niitä keskitetysti graafisen käyttöliittymän tai komentokehotteen avulla (Openstack Foundation 2018a, viitattu 29.3.2018).

2 PILVIPALVELUT

Pilvipalveluilla tarkoitetaan tietoliikenneverkon yli käytettäviä palveluita. Palvelut voivat olla muun muassa erilaisia sovelluksia, palvelimia, tallennustilaa tai alustoja verkkosivujen ylläpitämiseen. (Microsoft 2018a, viitattu 20.3.2018.) Palvelut mahdollistavat sen, että niiden käyttäjä voi esimerkiksi suorittaa paljon laskentatehoa vaativia ohjelmia tai ylläpitää omia verkkosivujaan ilman ylimääräisiä paikallisia resursseja. Tässä tapauksessa palvelun toimittaja vastaa näistä resursseista ja käyttäjä vain maksaa niiden käytöstä. Esimerkkinä pidetään tallennustilaa tarjoavia toimittajia. Kyseisessä tapauksessa palvelun tilaaja valitsee tarvittavan tallennustilan määrän ja palvelun toimittaja valmistelee sen tilaajan käyttöön. Tilaaajan ei tässä tapauksessa tarvitse huolehtia muusta kuin palvelun maksamisesta ja mahdollisesta tietoverkkoyhteydestä toimittajan tarjoamaan palveluun.

Pilvipalveluiden käytöstä maksetaan niiden käytön perusteella. Hyvänä puolena tässä periaatteessa on se, että hukkaan maksettuja resursseja ei synny ja kaikki maksetut palvelut käytetään yleensä hyödyksi. (Techtarget 2015, viitattu 20.3.2018.) Yrityksillä voi tietenkin olla myös erilaisia sopimuksia pilvipalveluiden toimittajien kanssa. Sopimus voi esimerkiksi kattaa tietyn määrän palveluita kuukautta kohden, ja ylimenevästä määrästä yritys maksaa hinnaston mukaisen maksun. (Microsoft 2018b, viitattu 28.4.2018.)

Pilvipalvelut voidaan jakaa kolmeen pilvityyppiin: julkisiin, yksityisiin ja hybrideihin. Julkisella pilvellä tarkoitetaan toimittajan ylläpitämää pilveä, jonka käytöstä yritys maksaa. Yksityisellä pilvellä tarkoitetaan yrityksen itse hallinnoivaa ja pystyttämää pilveä, joka useimmiten sijaitsee yrityksen omassa verkossa. Hybridipilvi taas nimensä mukaisesti yhdistelee julkisen sekä yksityisen pilven tarjoamia ominaisuuksia ja mahdollistaa tietojen jakamisen niiden kesken. (Microsoft 2018a, viitattu 20.3.2018.)

2.1 Julkinen pilvi

Julkisella pilvellä tarkoitetaan toimittajan ylläpitämää palvelukokonaisuutta, jonka käytöstä yritykset maksavat. Toimittaja vastaa pilven ylläpitämisestä laitteiston sekä tarjottavien palveluiden osalta. Julkisissa pilvissä käytössä olevat fyysiset resurssit eivät kohdistu pelkästään yhdelle asiakkaalle,

vaan resurssit on jaettu useamman asiakkaan kesken. Periaatteessa tämä tarkoittaa sitä, että samaa julkista pilveä käyttävät yritykset jakavat pilvelle varatut resurssit, mutta ovat kuitenkin eristettyinä toisistaan. (Snedeker 2017, viitattu 24.4.2018.)

Hyviä puolia julkisten pilvien käytössä on muun muassa se, että niiden avulla yritykset voivat saada helposti käyttöönsä lisää resursseja. Nykypäivänä datakeskukset vievät huomattavasti tilaa ja niiden ylläpitäminen ei välttämättä ole helppoa. Datakeskuksen päivittäminen voi olla todella hankalaa esimerkiksi tilan puutteen takia eivätkä niiden tarjoamat resurssit välttämättä vastaa yrityksen tarpeita. Julkista pilveä voidaan tässä tapauksessa pitää oikotienä hetkellisille lisäresursseille, koska pilvi skaalautuu yrityksen tarpeiden mukaan (Salo 2012, 23).

Julkisen pilven käyttöön liittyy tietenkin myös omat riskinsä. Julkista pilveä käyttävä yritys ei mahdollisesti tiedä missä pilveä ylläpitävä laitteisto sijaitsee tai miten sen tekninen puoli on toteutettu. Yritykset eivät saa säilyttää henkilötietojaan EU-alueen ulkopuolella lakien vuoksi. Lisäksi julkista pilveä käyttävällä yrityksellä saattaa herätä kysymys siitä, että onko palvelu aina saatavilla. (Salo 2012, 36 - 37.)

2.2 Yksityinen pilvi

Yksityinen pilvi tarjoaa samat ominaisuudet julkisen pilveen verrattuna muun muassa skaalautumisen osalta, mutta pilven kapasiteetti riippuu yrityksen omista resursseista. Yksityistä ja julkista pilveä verrattuna toisiinsa julkisen pilven toimittaja tarjoaa palveluitaan useille eri yrityksille. Yksityinen pilvi on taas useimmiten vain yhden yrityksen käytössä. Yksityinen pilvi on siis räätälöity ainoastaan vastaamaan yhden yrityksen tarpeita. (Interroute 2018, viitattu 26.3.2018.) Yksityiset pilvet perustuvat joko kaupallisiin tai avoimen lähdekoodin ratkaisuihin (Salo 2012, 27). Esimerkiksi Microsoftin Azure Stack -palvelu on yksi näistä kaupallisista ratkaisuista. (Microsoft 2018d, viitattu 16.5.2018).

Yksityiset pilvet sijaitsevat useimmiten yrityksen omissa tiloissa. Yritys täten tietää missä pilveä ylläpitämä laitteisto sijaitsee sekä kenellä on oikeus päästä siihen käsiksi. (Salo 2012, 28.) Yksityistä pilveä käyttävä yritys vastaa myös omasta palomuuristaan ja voi estää pilven käytön ulkopuolisista verkoista (Microsoft 2018c, viitattu 25.4.2018).

Yksityisen pilven käyttö ei juurikaan eroa julkisen pilven käytöstä. Yritykset tarvitsevat lisää henkilökuntaa pilven konfigurointia sekä ylläpitoa varten oli kyseessä taas yksityisen tai julkisen pilven käyttöönotto. Pilven käyttökustannukset muodostuvat tällöin tarvittavan laitteiston hankinnasta, konfiguroinnista, tuesta sekä ylläpitokustannuksista. Julkisten pilvien osalta toimittaja vastaa laitteiston hankinnasta ja sen ylläpitämisestä. (Techtarget 2017, viitattu 26.3.2018.)

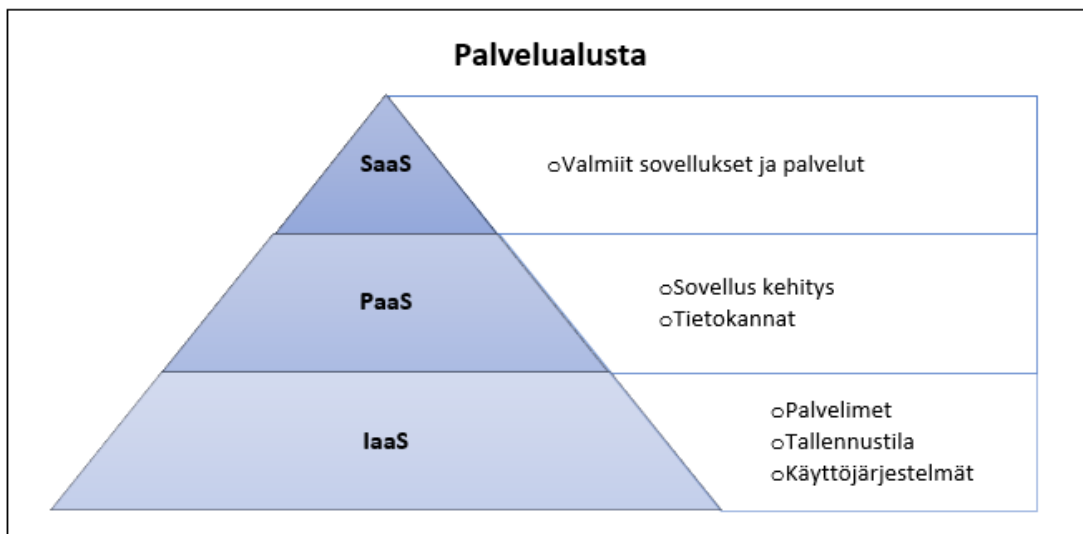
Yksityistä pilveä ei ole kuitenkaan pakko konfiguroida itse. Esimerkiksi yritykset, kuten RedHat sekä Rackspace, tarjoavat palvelunaan muun muassa yksityisen Openstack-pilven konfigurointia. Pilvi voi sijaita joko Rackspacen, kolmannen osapuolen tai yrityksen omissa tiloissa. (Rackspace 2018, viitattu 26.3.2018.)

2.3 Hybridipilvi

Hybridipilvet yhdistelevät julkisen sekä yksityisen pilven tarjoamia ominaisuuksia. Julkiset ja yksityiset pilvet muodostavat yhdessä kokonaisuuden, joka mahdollistaa tietojen ja sovellusten jakamisen pilvien kesken. Hybridipilvi mahdollistaa julkisen sekä yksityisen pilven etujen hyödyntämisen yhdessä. Yritys voi esimerkiksi säilyttää tärkeitä tietojansa yksityisen pilven puolella, mutta ajaa tietoja käyttävää sovellusta julkisen pilven toimittajan tarjoaman ratkaisun avulla. Hybridipilven avulla yritysten on myös mahdollista ohjata osa käyttämänsä sovelluksen kapasiteetista julkiseen pilveen ruuhka-aikoina. Tällöin yritykset maksavat ainoastaan satunnaisia kustannuksia julkisen pilven käytöstä eikä uuteen infrastruktuuriin ole pakko investoida pysyvästi. (Citrix 2014, viitattu 25.4.2018.)

3 PALVELUTYYPIT

Pilvipalvelut voidaan pilvityyppien lisäksi jakaa kolmeen palvelutyyppiin. Nämä tyypit ovat infrastruktuuri palveluna (IaaS), sovellus palveluna (SaaS) sekä alusta palveluna (PaaS). Kyseiset palvelutyyppit tarjoavat niitä käyttävällä yritykselle mahdollisuuden valita kuinka laajasta kokonaisuudesta yritys itse haluaa vastata omien tarpeidensa mukaan. Joskus mainittuja palvelutyypppejä kutsutaan pilvipinoksi, koska ne rakentuvat toistensa päälle (kuvio 1).



Kuvio 1. Palvelutyyppit ja esimerkkejä niiden käyttökohteista (Dusablon 2017, viitattu 26.4.2018)

3.1 Infrastruktuuri palveluna

IaaS-palvelut koostuvat toimittajan tarjoamasta perusrakenteesta. Toimittaja tarjoaa palveluita käyttäville yrityksille valmiin pohjan ja resurssit muun muassa palvelimien sekä muiden käyttöjärjestelmien tai ohjelmistojen hallintaan. IaaS-palveluiden avulla yritykset hallitsevat itse käyttämänsä ympäristöä ja pääsevät siihen kiinni aina tarpeen vaatiessa. (Salo 2012, 23.) Amazon ja Microsoft ovat suurimpia IaaS-palveluiden toimittajia (Dignan 2018, viitattu 16.5.2018).

IaaS-palveluiden hyviä puolia ovat niiden skaalautuvuus sekä joustavuus. Skaalautuvuudesta sekä joustavuudesta on hyötyä muun muassa erilaisissa testaustilanteissa palvelimen tai ohjelmistojen osalla. Toisena hyvänä puolena palveluiden tarjoamana etuna ovat mahdolliset rahalliset säästöt, jotka ilmaantuvat laitteiston ylläpitokustannusten hävitessä. (StateTech 2017, viitattu 28.3.2018.)

3.2 Alusta palveluna

PaaS-palvelut tarjoavat niitä käyttäville yrityksille alustan sovellusten kehittämiseen ja ylläpitämiseen. Palvelu ei korvaa kokonaista infrastruktuuria vaan tarjoaa avaimet erilaisten sovellusten kehittämiseen sekä integraatiomahdollisuuden tietokantojen kanssa. (Joshi 2014, viitattu 28.3.2018.) PaaS-palveluiden toimittajia ovat esimerkiksi Google, Microsoft, Amazon sekä Salesforce (Heino 2010, 51).

PaaS-palveluiden käyttö on yrityksille kustannustehokasta sekä niiden käyttö voi johtaa sovellusten nopeampaan kehittämiseen. Palveluiden avulla tuotettu lopputulos skaalautuu lähes loputtomasti käyttömäärän mukaan, jolloin turhaa aikaa ei tarvitse käyttää omaan infrastruktuuriin investoimiseen. (Salo 2012, 24.)

3.3 Sovellus palveluna

SaaS-palvelut tarjoavat yrityksille mahdollisuuden erilaisten sovellusten käyttöön tietoliikenneverkon yli toimittajan tarjoaman palvelun kautta (Heino 2010, 53). Yleisimpiä esimerkkejä tästä ovat sähköpostiohjelmat ja työväliohjelmistot, kuten Office365 tai Google Documents. Useat peruskäyttäjälle suunnatut Internetistä löytyvät SaaS-palvelut ovat ilmaisia, mutta yrityskäyttöön räätälöidyistä sovelluksista niitä käyttävä yritys maksaa vuokraa. Yrityskäyttöön räätälöityä sovelluksia ovat esimerkiksi Salesforcen tarjoamat taloushallinnon sovellukset (Salesforce 2018, viitattu 26.4.2018).

Hyöty SaaS-palveluiden käytössä on niiden saatavuus. Sovellukset ovat saatavilla paikoissa, joissa on verkkoyhteys, sekä niiden käyttö onnistuu yleensä verkkoselaimen kautta. Toinen hyvä puoli niiden käytössä käyttäjäkohtaisesti on, että sovellusta ei tarvitse itse päivittää. (Chrzanowska 2016, viitattu 28.3.2018.)

4 OPENSTACK-PILVIPALVELUALUSTA

Openstack-pilvipalvelualusta on avoimeen lähdekoodiin perustuva kokoelma palveluja pilvipalvelualustojen hallintaan ja rakentamiseen. Palveluja kutsutaan projekteiksi. Openstack voidaan luokitella infrastruktuuri palveluna -ratkaisuksi. Jokainen Openstack-alustan palvelu tarjoaa API-ohjelmointirajapinnan (Application Programmin Interface). Openstack on voittoa tavoittelemattoman Openstack-säätiön ylläpitämä. Säätiö valvoo Openstack-alustan kehitystä ja yhteisön rakentumista sen ympärillä. (Opensource 2018, viitattu 29.3.2018.) Sadat maailmanlaajuiset yritykset, kuten Ebay, käyttävät Openstack-alustaa apuna päivittäisen liiketoiminnan tukena (Openstack Foundation 2018a, viitattu 29.3.2018).

Openstack syntyi kahden toimijan yhteistyönä vuoden 2010 alussa. Nämä toimijat olivat Rackspace ja Nasalle työskentelevä Anso Labs. Rackspace halusi uudelleen koodata pilvipalvelualustansa ja tehdä jo valmiista koodista avoimen muille käyttäjille. Samoihin aikoihin Nasalle työskentelevä Anso Labs oli julkaissut alustavan koodin Nova-palvelulle, joka on yksi Openstack-alustan ydinprojekteista. Nova on Openstack-alustan compute-palvelu. Osapuolet ottivat yhteyttä toisiinsa ja loivat perustan Openstack-alustalle. Vuonna 2012 Openstack-säätiö perustettiin yksityisenä tahona Openstackin resurssien jakamiseen sekä yhteisön mainostamista varten. (Openstack Foundation 2018b, viitattu 29.3.2018.)

Tällä hetkellä Openstack koostuu kaiken kaikkiaan noin 40 eri projektista. Lähes kaikkia näitä projekteja yhdistää kuuden kuukauden kehityssykli. Openstack pyrkii täten julkaisemaan kaksi kertaa vuodessa uuden version alustastaan. (Openstack Foundation 2018a, viitattu 29.3.2018). Openstackin viimeisin uusi versio koodinimeltään Queens julkaistiin helmikuussa 2018. Taulukosta 1 käyvät ilmi Openstackin eri versiot sekä niiden julkaisupäivät.

Taulukko 1. Openstackin eri versiot ja niiden julkaisu päivät (Openstack Foundation 2018c, viitattu 29.3.2018).

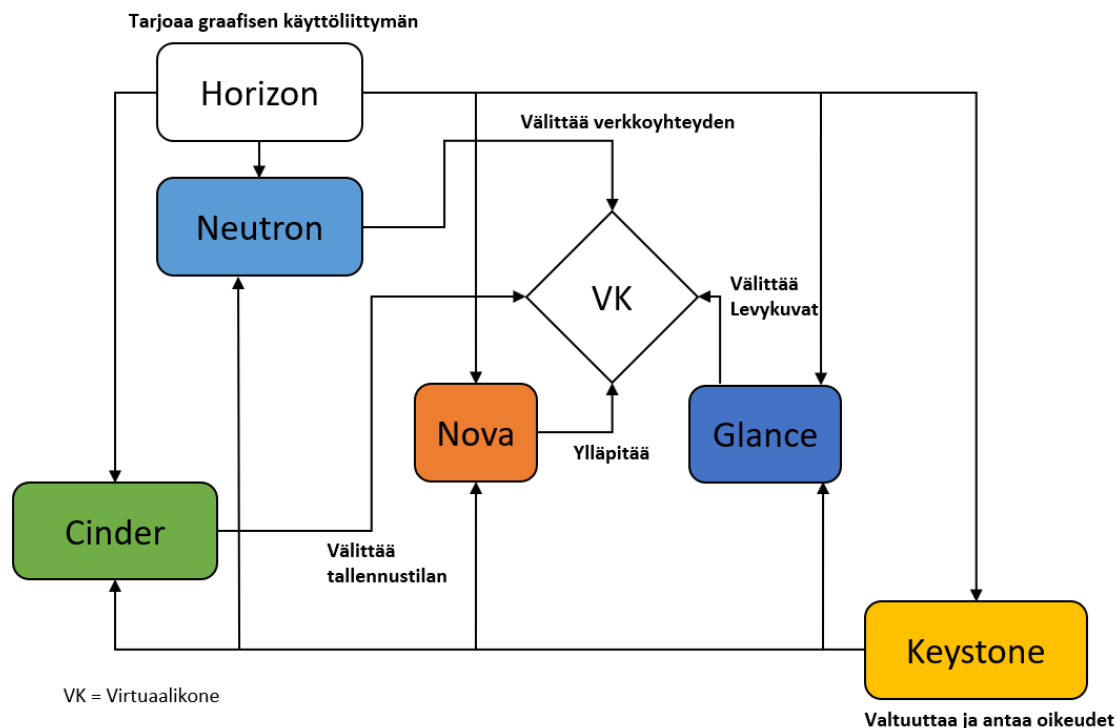
Koodinimi	Julkaisu ajankohta
Austin	21.10.2010
Bexar	03.02.2011
Cactus	15.04.2011
Diablo	22.09.2011
Essex	05.04.2012
Folsom	27.09.2012
Grizzly	04.04.2013
Havana	17.10.2013
Icehouse	17.04.2014
Juno	16.10.2014
Kilo	30.04.2015
Liberty	15.10.2015
Mitaka	07.04.2016
Netwon	06.10.2016
Ocata	22.02.2017
Pike	30.08.2017
Queens	28.02.2018
Rocky	Syksy 2018

4.1 Testiympäristö

Openstack-ympäristö muodostuu käytännössä kahdesta tai useammasta isäntätietokoneesta. Isäntätietokoneita Openstack kutsuu termillä noodi. Yrityskäytössä Openstack asennetaan fyysisille isäntätietokoneille riittävän suorituskyvyn takaamiseksi. Testitarkoituksessa Openstack voidaan asentaa myös virtuaalikoneille, mutta yrityskäytössä tämä ei ole suositeltavaa. Virtuaalikoneille asentaminen esimerkiksi alentaa Openstack-alustan suorituskykyä. (Openstack Foundation 2018x, viitattu 24.4.2018.)

Openstackin testiasennuksen avulla perehdytään Openstackin toimintaan pilvipalvelualustana. Asennus antaa paremman kuvan ympäristön rakenteen ja hallinnan osalta verrattuna valmiin kokonaisuuden tutkimiseen. Testiasennuksen tavoitteena on tutustua Openstackin eri palveluiden toimintaan ja konfiguroida toimiva Openstack-ympäristö, jonka käyttöä testataan.

Rakenteeltaan testiympäristö koostuu useista palveluista. Jokaista palvelua pidetään omana projektinaan. Palveluja voidaan siis kehittää itsenäisesti muiden palveluiden tilasta riippumatta. (Openstack Foundation 2018ä, viitattu 4.5.2018.) Openstackin Keystone-palvelua voidaan pitää testiympäristön tärkeimpänä palveluna. Muun muassa suurin osa testiympäristöön kuuluvista palveluista tarvitsee Keystone-palvelun antamia oikeuksia toimiakseen. Kuviosta 2 käy ilmi, kuinka Openstack-ympäristön palvelut toimivat yhdessä muodostaessaan Openstack-pilvipalvelualustan.

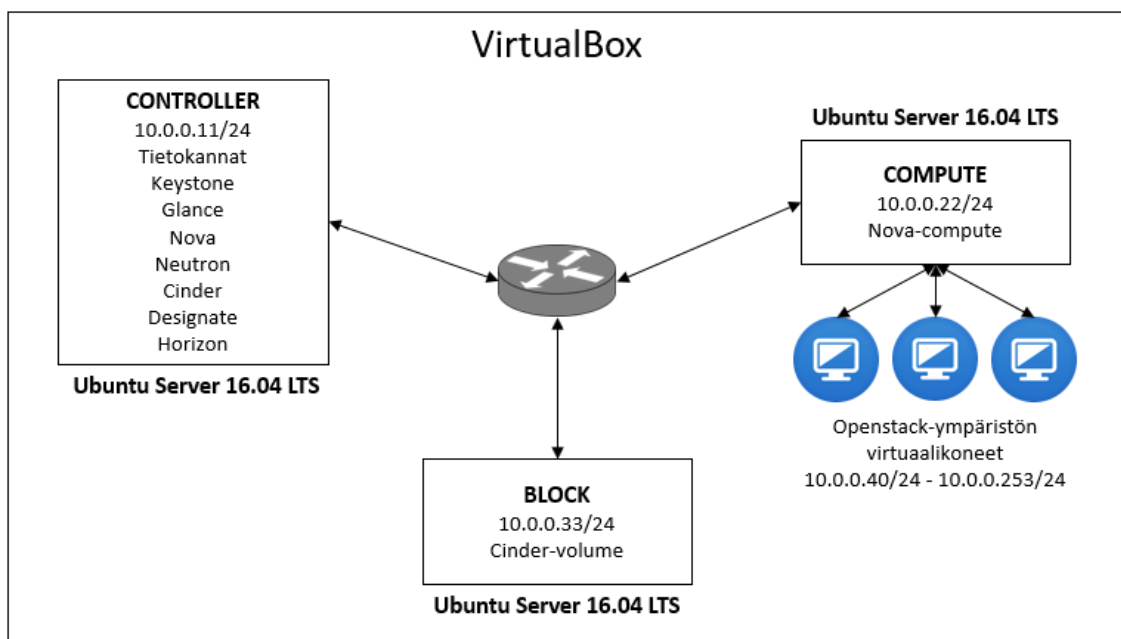


Kuvio 2. Palveluiden yhteydet (Openstack Foundation 2018ö, viitattu 4.5.2018)

Periaatteessa jokainen testiympäristön palvelu voidaan asentaa samalle isäntätietokoneelle, mutta käytännössä se ei ole rahallisesti suositeltavaa. Käytännössä useita isäntätietokoneita, noodeja, käyttämällä Openstack-ympäristöstä saadaan enemmän tehoa irti, ja sen laajentaminen onnistuu helposti muun muassa uusien noodien lisäyksellä ympäristöön.

Openstack-ympäristön minimaalinen asennus koostuu neljästä eri palvelusta jotka ovat identiteetipalvelu Keystone, levykuvapalvelu Glance, compute-palvelu Nova sekä verkkopalvelu Neutron. Periaatteessa kyseiset neljä palvelua riittävät Openstack-ympäristön asentamiseen, mutta näiden palveluiden lisäksi Openstack suosittelee vielä asennettavaksi graafisen käyttöliittymän sekä Cinder-varastointipalvelun. (Openstack Foundation 2018d, viitattu 2.4.2018.)

Tässä opinnäytetyössä konfiguroitava Openstack-pilvipalvelualustan testiympäristö muodostuu kolmesta VirtualBoxilla ajettavasta virtualisoidusta isäntätietokoneesta, joille edellä mainitut palvelut sekä graafinen käyttöliittymä Horizon ja DNS-palvelu Designate asennetaan. Isäntätietokoneet nimetään seuraavasti: Controller, Compute ja Block. Kaikille isäntätietokoneille, noodeille, tehtävät alkuvalmistelut ovat samankaltaiset, mutta niille asennettavat palvelut eroavat toisistaan niiden yksilöimiseksi. Controller-noodin tehtävänä on keskittyä koko ympäristön ylläpitämiseen ja hallintaan. Compute-noodi on ympäristön hypervisor ja tarjoaa minimaaliset palvelut virtuaalikoneiden hallintaan. Hypervisor tarkoittaa isäntätietokoneessa suoritettavaa ohjelmaa, jonka avulla toteutetaan yhden tai useamman käyttöjärjestelmän tarvitsemat palvelut isäntätietokoneen päällä oleville virtuaalikoneille (Heino 2010, 258). Block-noodin tehtävänä on pitää huolta testiympäristössä ajettavien virtuaalikoneiden tarvitsemasta levytilasta. Kolmen noodin muodostama testiympäristö simuloi sitä, kuinka Openstack tultaisiin käytännössä asentamaan yrityskäytössä. Kuviosta 3 käy ilmi asennettavan testiympäristön rakenne ja tärkeimmät noodeille asennettavat palvelut. Asennettava Openstack-alustan versio on Pike.



Kuvio 3. Testiympäristön rakenne

4.2 Alkuvalmistelut

Openstack asennetaan virtuaalikoneille, koska testiympäristön tavoitteena on saada pystytettyä infrastruktuuri virtuaalikoneiden hallintaan. Ympäristö ei täten vaadi valtavia resursseja, koska ympäristöä käytetään apuna Openstack-alustan toimintaan perehdyttäessä. Openstack asennetaan Openstack-säätiön dokumentoinnin perusteella kolmelle virtualisoidulle isäntätietokoneelle. (Openstack Foundation 2018a, viitattu 29.3.2018). Ympäristön rakentaminen aloitetaan isäntätietokoneiden, noodien, käyttöjärjestelmien asennuksella. Noodeille asennettava käyttöjärjestelmä on Ubuntu Server 16.04 LTS. Jokaiselle noodille määritetään kiinteä IP-osoite. Noodien IP-osoitteet näkyvät kuviossa 3. Jokaisen noodin `/etc/hosts`-tiedostoa muokataan nimenselvitystä varten vastaamaan noodien IP-osoitteita (kuvio 4).

```
127.0.0.1      localhost
10.0.0.11      controller
10.0.0.22      compute
10.0.0.33      block
...
```

Kuvio 4. Noodien `etc/hosts`-tiedoston muokkaus

Käyttöjärjestelmän asentamisen ja päivittämisen jälkeen noodeille asennetaan Chrony NTPn (Network Time Protocol) käyttöä varten. Chrony-palvelun tavoitteena on synkronoida noodien kello julkisen NTP-palvelimen kanssa, jolloin jokaiselle noodille saadaan yhtenäinen aika luotettavuuden ja toiminnan parantamiseksi (Chrony 2017, viitattu 26.4.2018). Chrony-palvelu asennetaan komennolla `"apt install chrony"`. Controller-noodin konfigurointitiedostoa `/etc/chrony/chrony.conf` muokataan ja lisätään seuraava rivi (kuvio 5).

```
server fi.pool.ntp.org iburst
allow 10.0.0.0/24
...
```

Kuvio 5. `Chrony.conf`-tiedoston muokkaus Controller-noodilla

Tämän jälkeen Chrony-palvelu käynnistetään uudelleen komennolla `"service chrony restart"`. Chrony-palvelun asentaminen muille noodeille suoritetaan samalla tavalla, mutta konfigurointitiedostoa muokataan viittaamaan Controller-noodiin (kuvio 6), jonka jälkeen Chrony-palvelu käynnistetään uudelleen.


```
server controller iburst
---
```

Kuvio 6. Chrony.conf-tiedoston muokkaus muilla noodeilla

Chrony-palvelun asennuksen jälkeen noodeilla otetaan käyttöön Openstack-alustan tarvitsemat pakettivarastot. Tämä tapahtuu komennoilla "apt install software-properties-common" ja "add-apt-repository cloud-archive:pike". Pakettivarastojen käyttöönoton jälkeen suoritetaan vielä komento "apt update && apt dist-upgrade" pakettien päivittämiseksi. Lopuksi asennetaan Openstack-alustan asiakasohjelma komennolla "apt install python-openstackclient" Openstack-alustan omien komentojen käyttöä varten.

Seuraavaksi Controller-noodille asennetaan SQL-tietokanta, koska useat Openstack-alustan palveluista tarvitsevat tietokannan tietojen säilyttämiseen. Testiympäristössä käytettävä tietokanta on MariaDB, mutta Openstack tukee myös muita vaihtoehtoja. MariaDB-tietokannan asentaminen aloitetaan komennolla "apt install mariadb-server python-pymysql". Asennuksen jälkeen luodaan tiedosto *99-openstack.cnf* sijaintiin */etc/mysql/mariadb.conf.d/99-openstack.cnf*. Tiedostoon lisätään seuraavat rivit (kuvio 7).

```
[mysqld]
bind-address = 10.0.0.11

default-storage-engine = innodb
innodb_file_per_table = on
max_connections = 4096
collation-server = utf8_general_ci
character-set-server = utf8
```

Kuvio 7. 99-openstack.cnf-tiedoston sisältö

Bind-address on tässä tapauksessa Controller-noodin IP-osoite. Osoite on määritettävä, jotta muutkin noodit pääsevät tietokantaan kiinni. Tiedoston muokkauksen jälkeen tietokantapalvelu käynnistetään uudelleen komennolla "service mysql restart". Palvelun uudelleenkäynnistyksen jälkeen suoritetaan ohjelmakoodi "mysql_secure_installation", jossa määritetään muun muassa salaisana tietokannan root-käyttäjälle.

Tietokannan valmistelun jälkeen Controller-noodille asennetaan viestijonopalvelu RabbitMQ. Openstack käyttää viestijonopalvelua toimenpiteiden sekä palveluiden tilojen koordinoimiseen. RabbitMQ-palvelu asennetaan komennolla "apt install rabbitmq-server". Asennuksen jälkeen

viestijonopalvelun avulla luodaan käyttäjä Openstack komennolla `"rabbitmqctl add_user openstack salasana"`. Luonnin jälkeen käyttäjällä annetaan oikeudet hallintaan, kirjoitukseen ja lukuun komennolla `"rabbitmqctl set_permissions openstack \".*\" \".*\" \".*\""`.

Seuraavaksi Controller-noodille asennetaan Memcached-palvelu. Openstack-alustan identiteettipalvelun todennusmekanismi käyttää Memcached-palvelua valtuuksien kätkö paikkana. Memcached-palvelu asennetaan komennolla `"apt install memcached python-memcache"`. Asennuksen jälkeen muokataan tiedostoa `/etc/memcached.conf` vastaamaan Controller-noodin IP-osoitetta, jotta muutkin noodit saavat yhteyden palveluun. Tiedoston muokkauksen jälkeen palvelu käynnistetään uudelleen komennolla `"service memcached restart"`.

4.2.1 Keystone

Keystone on Openstack-alustan identiteettipalvelu. Keystone-palvelu on asennettava ensimmäisenä testiympäristöä rakentaessa. Palvelua käytetään muun muassa käyttäjien todennuksen ja valtuuksien hallintaan. Hallinnan lisäksi Keystone-palvelu ylläpitää palveluluetteloa joka on kokoelma saatavilla olevista palveluista Openstack-ympäristössä. (Openstack Foundation 2018e, viitattu 3.4.2018.) Identiteettipalvelun hallitsema kokonaisuus koostuu toimialueista, projekteista, ryhmistä ja käyttäjistä. Toimialueet ovat korkeimman tason säiliöitä edellä mainitulle kohteille. Projektit taas muodostavat perusyksikön resurssien omistamiselle. Openstack-ympäristössä kaikkien resurssien pitäisi olla jonkun projektin omistamia ja projektin pitäisi olla tietyn toimialueen omistama. Sama rakenne toistuu ryhmien ja käyttäjien kohdalla. Tämä tarkoittaa sitä, että käyttäjät, ryhmät sekä projektit ovat ainutlaatuisia ainoastaan toimialuekohtaisesti. (Openstack Foundation 2018f, viitattu 3.4.2018.)

Keystone-palvelu käyttää Fernet-valtuuksia käyttäjien todennukseen ja valtuuttamiseen. Valtuudet sisältävät minimaaliset tiedot edellä mainittujen toimintojen toteuttamiseen. Normaalissa tapauksessa valtuus sisältää ainutlaatuisen käyttäjätunnuksen sekä ainutlaatuisen projektitunnuksen. Näiden lisäksi valtuus sisältää luontiajan, valtuuden eliniän sekä todennustavat. (Dolph 2015, viitattu 3.4.2018.)

Keystone-palvelun asennus aloitetaan tietokannan luomisella. Tietokannan luonti aloitetaan kirjautumalla tietokantapalveluun root-käyttäjänä komennolla `"mysql"`. Kirjautumisen jälkeen

luodaan Keystone niminen tietokanta, jolle määritetään tarvittavat käyttöoikeudet sekä salasana (liite 1). Tietokannan luonnin jälkeen Keystone-palvelu voidaan asentaa komennolla "apt install keystone apache2 libapache2-mod-wsgi". Komento asentaa identiteettipalvelun lisäksi Apache http-palvelimen sekä mod-wsgi-moduulin, joita käytetään identiteettipalvelun pyyntöjen hoitamiseen portissa 5000. Paketit hoitavat automaattisesti Apache-palvelimen konfiguroinnin sekä mod-wsgi-moduulin aktivoinnin. Identiteettipalvelun asennuksen jälkeen muokataan sen konfigurointitiedostoa `/etc/keystone/keystone.conf` vastaamaan luomaamme tietokantaa ja määritetään valtuuksien toimittajaksi Fernet (kuvio 8).

```
[database]
...
connection = mysql+pymysql://keystone:keystone@controller/keystone

[token]
...
provider = fernet
```

Kuvio 8. Keystone.conf-tiedoston muokkaus

Konfigurointitiedoston muokkauksen jälkeen tietokanta täytetään identiteettipalvelun tiedoilla komennolla "su -s /bin/sh -c "keystone-manage db_sync" keystone". Seuraavaksi Fernet-avainten arkistot alustetaan ja identiteettipalveluun luodaan salasana admin-tunnukselle sekä loppupisteet admin-, internal- sekä public-liitännöille (liite 1). Openstack käyttää loppupisteitä kommunikointiin muiden Openstack-ympäristön palveluiden kanssa. Tämän jälkeen Apache-palvelimen konfigurointitiedoston `/etc/apache2/apache2.conf` loppuun lisätään rivi kuvaamaan palvelinnimen olevan Controller.

Seuraavaksi Apache-palvelimen palvelut käynnistetään uudelleen komennolla "service apache2 restart" ja luodaan tiedosto admin-openrc (liite 1) Openstack-alustan käytön helpottamiseksi. Openstack-alustan omien kommentojen käyttö vaatii admin-tunnuksen kirjautumistietojen lataamisen useamman eri komennon kautta ja hidastaa näin sen käyttöä. Luodun tiedoston avulla kirjautumistietojen lataaminen onnistuu yksinkertaisesti komennolla ". admin-openrc". Kirjautumistietojen lataamisen jälkeen luodaan ympäristön eri palveluiden admin-käyttäjille projekti komennolla "openstack project create --domain default --description "Service Project" service" identiteettipalvelun asennuksen viimeistelyksi. (Openstack Foundation 2018j, viitattu 6.4.2018.)

4.2.2 Glance

Glance on Openstack-alustan levykuvapalvelu. Palvelu tarjoaa käyttäjälleen mahdollisuuden virtuaalikoneiden levykuvien metatietojen havaitsemiseen, rekisteröimiseen sekä kokonaisten levykuvien noutamiseen. Glance-palvelun levykuvien tallentaminen onnistuu paikallisille kiintolevyille tai Openstack-alustan omaan objektivarastoon. (Openstack Foundation 2018g, viitattu 4.4.2018.) Glance-palvelun levykuvat varastoidaan mallipohjina, joita käytetään uusien virtuaalikoneiden pysyttämiseen. Levykuvia Glance-palveluun ladatassa käyttäjän täytyy määritellä minkä tyyppinen levykuva on. Levykuvapalvelun tukemia levymuotoja ovat muun muassa VHD, VMDK sekä qcow2. Näiden lisäksi Glance-palvelu tukee myös säiliömuotoa. Säiliömuoto kertoo tyyppin lisäksi lisätietoja itse virtuaalikoneesta. (Shameen 2016, viitattu 4.4.2018). Yksi näistä tuettavista muodoista on AMI (Amazon Machine Image), joka viittaa Amazon Web Services -pilvipalvelun käyttämään tiedostotyyppiin.

Levykuvapalvelun asennus aloitetaan identiteettipalvelun tavoin tietokannan luomisella. Tietokanta luodaan ja sille määritellään tarvittavat oikeudet sekä salasana. Tietokannan luomisen jälkeen ladataan admin-tunnuksen kirjautumistiedot komennolla `"admin-openrc"` Openstack-alustan komentojen käyttämiseksi. Tämän jälkeen luodaan käyttäjä Glance komennolla `"openstack user create --domain default --password-prompt glance"`. Komento luo käyttäjän ja samalla sille määritetään salasana. Luonnin jälkeen käyttäjälle lisätään admin-rooli komennolla `"openstack role add --project service --user glance admin"`. Seuravaksi luodaan Glance-palvelukokonaisuus komennolla `"openstack service create --name glance --description "OpenStack Image" image"` sekä loppupisteet levykuvapalvelulle (Liite 2). Komennon suorittamisen jälkeen identiteettipalvelu palauttaa vastaavan ikkunan, josta käy ilmi muun muassa palvelun saama tunniste (kuvio 9).

Field	Value
description	OpenStack Image
enabled	True
id	e65ed6a8e5de43769aa6fbb6058308fb
name	glance
type	image

Kuvio 9. Openstack-alustan levykuvapalvelun luonti

Seuraavaksi levykuvapalvelu asennetaan komennolla `"apt install glance"`. Levykuvapalvelun konfigurointitiedostoa `/etc/glance/glance-api.conf` muokataan (Liite 2). Konfigurointitiedostossa määritetään Glance-palvelu käyttämään luotua tietokantaa sekä yhteys identiteettipalveluun. Lisäksi määritellään levykuvapalvelun käyttämä tiedostosijainti palveluun ladatuille levykuville. Samat muutokset konfiguroidaan myös tiedostoon `/etc/glance/glance-registry.conf` lukuun ottamatta levykuvien tiedostosijaintia. Lopuksi levykuvapalvelun tietokanta täytetään komennolla `"su -s /bin/sh -c "glance-manage db_sync" glance"` ja palvelut käynnistetään uudelleen komennoilla `"service glance-registry restart"` ja `"service glance-api restart"`.

Levykuvapalvelun toiminnan testaamiseksi testiympäristöön ladataan levykuva (Liite 3). Lataukseen käytettävä komento määrittää käytettävän levytyypin, säiliötyypin sekä sen näkyvyyden. Tässä tapauksessa levykuva on määritetty julkiseksi, jotta kaikki Openstack-ympäristömme käyttäjät pääsevät siihen käsiksi. (Openstack Foundation 2018k, viitattu 7.4.2018.)

4.2.3 Nova

Nova on Openstack-ympäristön Compute-palvelu, jonka tehtävänä on hallinnoida ja ylläpitää ympäristön virtuaalikoneita. Nova vaatii toimiakseen vähintään Keystone-palvelun, Glance-palvelun sekä Neutron-palvelun tarjoamat palvelut. Nova muodostuu useista eri palveluprosesseista, joista jokainen suorittaa erilaisia funktioita. (Openstack Foundation 2018h, viitattu 11.4.2018.) Suurin osa Nova-palveluun kuuluvista palveluista asennetaan Controller-noodille, mutta itse Compute-palvelu asennetaan omalle noodilleen.

Nova-palvelun asentaminen aloitetaan tietokantojen luomisella. Nova tarvitsee kaiken kaikkiaan kolme eri tietokantaa palveluilleen. Luodaan siis kolme tietokantaa nimeltä `nova_api`, `nova` sekä `nova_cell0` ja annetaan niille tarvittavat oikeudet (liite 4). Tietokantojen valmistelun jälkeen ladataan admin-käyttäjän kirjautumistiedot komennolla `". admin-openrc"` ja luodaan käyttäjä Nova komennolla `"openstack user create --domain default --password-prompt nova"`. Luonnin jälkeen lisätään käyttäjälle vielä admin-rooli komennolla `"openstack role add --project service --user nova admin"`. Seuraavaksi luodaan Nova-palvelukokonaisuus komennolla `"openstack service create --name nova --description "OpenStack Compute" compute"`. Levykuvapalvelun tavoin compute-palvelulle luodaan loppupisteet `public-`, `internal-` sekä `admin-liitännöille`. (liite 4.)

Seuraavaksi luodaan käyttäjä nimeltä Placement. Käyttäjälle lisätään myös admin-rooli. Lisäksi Placement API -palvelulle luodaan palvelukokonaisuus komennolla "openstack service create --name placement --description "Placement API" placement". Nova käyttää Placement API -palvelua resurssien luetteloimiseen sekä käytön seurantaan. Resurssit voivat olla muun muassa erillisiä noodeja, jaettuja tallennustiloja tai IP-osoitteita. (Openstack Foundation 2018i, viitattu 11.4.2018.) Lopuksi Placement API -palvelulle luodaan loppupisteet public-, internal- sekä admin-liitäntöille (liite 4).

Seuraavaksi asennetaan Controller-noodille Nova-palvelun tarvitsemat paketit komennolla " apt install nova-api nova-conductor nova-consoleauth nova-novncproxy nova-scheduler nova-placement-api". Placement-palvelun lisäksi komento asentaa muun muassa tarvittavat palvelut tietokantoja sekä Nova-palvelun konsolia varten. Lisäksi Nova-scheduler-palvelun tehtävänä on vastata, minkä Compute-noodin päällä testiympäristössä luodut virtuaalikoneet tullaan ajamaan. Pakettien asennuksen jälkeen Nova-palvelun konfigurointitiedostoa */etc/nova/nova.conf* muokataan (liite 5). Konfigurointitiedossa määritellään muun muassa Nova-palvelun käyttämät tietokannat, yhteys identiteettipalveluun, Neutron-palvelun käyttö sekä Glance-palvelun osoite. Muutosten jälkeen Nova-palvelun tietokannat otetaan käyttöön sekä rekisteröidään alussa luotu cell0-tietokanta ja luodaan cell1-solu (liite 4). Lopuksi asennetut palvelut käynnistetään uudelleen. (Openstack Foundation 2018l, viitattu 11.4.2018.)

Controller-noodin asennuksen jälkeen Compute-noodille asennetaan Nova-compute-palvelu komennolla "apt install nova-compute". Compute-noodin konfigurointitiedostoa */etc/nova/nova.conf* muokataan Controller-noodin tavoin pienien muutosten kera (liite 5). Compute-noodin konfigurointitiedostosta jätetään tietokantayhteydet konfiguroimatta, koska itse compute-palvelu ei tarvitse suoraa yhteyttä mihinkään tietokantaan. Lisäksi IP-osoitteeksi määritetään Compute-noodin käyttämä IP-osoite sekä VNC-protokolla otetaan käyttöön. VNC-protokollan käytön avulla Compute-noodin päällä ajettavien virtuaalikoneiden graafisiin käyttöliittymiin pääsee käsiksi verkkoselaimen avulla. Konfigurointitiedoston muokkauksen jälkeen Compute-noodilla suoritetaan komento "egrep -c '(vmx|svm)' /proc/cpuinfo". Komennon palauttama arvo kertoo, että tukeeko Compute-noodi laitteistokiihdytystä. Komennon palauttama arvo on nolla, joka viittaa siihen, että Compute-noodi ei tue kyseistä ominaisuutta. Tämän vuoksi Nova-palvelun konfigurointitiedostoa on muokattava hieman ja käytettäväksi virtualisointityypiksi on määritettävä QEMU (liite 5). Muokkauksen jälkeen Nova-compute-palvelu käynnistetään uudelleen komennolla "service nova-compute restart".

Nova-palvelun toiminnan testaamiseksi Controller-noodilla suoritetaan komento "openstack compute service list". Komento listaa ympäristön compute-palvelut. Sen avulla voi varmistaa, että palvelut toimivat oikein (kuvio 10). (Openstack Foundation 2018m, viitattu 12.4.2018.)

```
root@controller:/home/sami# openstack compute service list
```

ID	Binary	Host	Zone	Status	State	Updated At
3	nova-consoleauth	controller	internal	enabled	up	2018-04-12T12:57:14.000000
4	nova-scheduler	controller	internal	enabled	up	2018-04-12T12:57:21.000000
5	nova-conductor	controller	internal	enabled	up	2018-04-12T12:57:14.000000
6	nova-compute	compute	nova	enabled	up	2018-04-12T12:57:23.000000

Kuvio 10. Nova-palvelun toiminnan testaus

4.2.4 Neutron

Neutron on Openstack-alustan verkkopalvelu, jonka tehtävänä on tarjota verkkoyhteys Openstack-alustan ylläpitämien virtuaalisten liitäntöjen välillä. Neutron-palvelun avulla voi luoda erilaisia verkkoja jotka voidaan kytkeä muun muassa Nova-palvelun ylläpitämiin virtuaalikoneisiin. (Openstack Foundation 2018o, viitattu 12.4.2018.) Neutron tukee kahta erilaista verkkovaihtoehtoa, jotka ovat toimittajaverkot sekä itsepalveluverkot. Toimittajaverkko on Neutron-palvelun yksinkertaisin tapa tarjota virtuaalikoneille verkkoyhteys. Kyseinen konfiguraatio mahdollistaa virtuaalisten verkkojen siltaamisen fyysisiin verkkoihin ja käyttää fyysisen verkon tarjoamaa reititysmahdollisuutta. Itsepalveluverkkojen konfiguraatio taas mahdollistaa virtuaalisten reitittimien luomisen sekä NATin (Network Address Translation) käytön virtuaalisten verkkojen reitityksessä fyysisen verkon kanssa. Verkkopalvelun konfigurointi eroaa hieman eri verkkovaihtoehtojen kesken. Testiympäristössä päädyttiin yksinkertaisempaan toimittajaverkkoon. (Openstack Foundation 2018p, viitattu 12.4.2018.)

Verkkopalvelun asentaminen aloitetaan Controller-noodilla tietokannan sekä käyttäjän luomisella. Seuraavaksi luodaan Neutron-palvelukokonaisuus komennolla "openstack service create --name neutron --description "Openstack Networking" network". Lisäksi verkkopalvelulle luodaan loppupisteet Openstack-alustan muiden palveluiden tavoin public-, internal- sekä admin-liitäntöille (liite 6). Seuraavaksi Controller-noodille asennetaan verkkopalvelun tarvitsemat paketit komennolla "apt install neutron-server neutron-plugin-ml2 neutron-linuxbridge-agent neutron-dhcp-agent neutron-metadata-agent". Komento asentaa tarvittavat paketit myös DHCP-palvelun käyttöä

varten IP-osoitteiden jakamiseksi ympäristöön luotaville virtuaalikoneille. Pakettien asennuksen jälkeen verkkopalvelun konfigurointitiedostoa */etc/neutron/neutron.conf* muokataan (liite 6). Konfigurointitiedostossa määritetään muun muassa verkkopalvelun tietokanta sekä otetaan käyttöön Modular Layer 2 -liitännäinen. Liitännäinen mahdollistaa virtuaalisten kytkimien käytön Openstack-ympäristössä.

Verkkopalvelun konfiguroinnin jälkeen ML2-liitännäisen konfigurointitiedostoa muokataan (liite 7). Konfigurointitiedostossa määritetään käytettäväksi verkkotyyppiä flat ja vlan. Flat viittaa tässä tapauksessa fyysiseen verkkoon. Lisäksi tiedostossa otetaan käyttöön Linux Bridge Agent -liitännäinen. Liitännäistä käytetään virtuaalisten verkkoinfrastruktuurien rakentamiseen. LBA-liitännäisen konfigurointitiedostoa */etc/neutron/plugins/ml2/linuxbridge_agent.ini* (liite 7) muokataan. Tiedostossa määritetään muun muassa Linux Bridgen Agent -liitännäisen käyttämä fyysinen liitäntä sekä turvallisuusryhmien käyttöönotto. Ryhmien avulla voi muun muassa määritellä, saavatko siihen kuuluvat virtuaalikoneet internetyhteyden tai sallitaanko niillä ping-komennon käyttö. Lisäksi muokataan tiedostoa */etc/neutron/dhcp_agent.ini* DHCP-palvelun käyttämiseksi testiympäristössämme (liite 7).

Seuraavaksi muokataan Metadata Agent -liitännäisen konfigurointitiedostoa sekä määritetään Nova käyttämään verkkopalvelun luotuja tietoja (liite 8). Openstack-ympäristön virtuaalikoneet käyttävät Metadata Agent -liitännäistä virtuaalikonekohtaisten tietojen noutamiseen. Tiedot voivat koskea virtuaalikoneen julkista IP-osoitetta, isäntänimeä tai SSH-avainta (Openstack Foundation 2018p, viitattu 12.4.2018). Controller-noodin asennuksen viimeistelyksi verkkopalvelun tietokannat otetaan käyttöön ja tarvittavat palvelut käynnistetään uudelleen (liite 8). (Openstack Foundation 2018q, viitattu 12.4.2018.)

Openstack-ympäristön jokaiselle Compute-noodille on asennettava verkkopalvelun Linux Bridge Agent -liitännäinen. Linux Bridge Agent -liitännäinen asennetaan Compute-noodille komennolla "apt install neutron-linuxbridge-agent". Asennuksen jälkeen Neutron-palvelun sekä Nova-palvelun konfigurointitiedostoa muokataan (liite 9). Lisäksi Linux Bridge Agent -liitännäisen konfigurointitiedostoa muokataan Controller-noodin tavoin (liite 7). Asennuksen viimeistelyksi Nova- sekä Neutron-palvelun palvelut käynnistetään uudelleen Compute-noodilla. Lopuksi verkkopalvelun toiminta varmistetaan suorittamalla komento "openstack neutron agent list" Controller-noodilla (kuvio 11). (Openstack Foundation 2018r, viitattu 12.4.2018.)


```
root@controller:/home/sami# openstack network agent list
```

ID	Agent Type	Host	Availability Zone	Alive	State
3a701474-2eda-4667-b8bf-	Linux bridge agent	compute	None	:-)	UP
735b211f-2a43-4ec1-907d	Linux bridge agent	controller	None	:-)	UP
a083f9f4-fe19-4b19-8bc9-	Metadata agent	controller	None	:-)	UP
d5accb65-41c5-4ef2-af85	DHCP agent	controller	nova	:-)	UP

Kuvio 11. Verkkopalvelun toiminnan testaus

4.2.5 Cinder

Cinder on Openstack-alustan varastointipalvelu, jonka tehtävänä on tarjota tallennustilaa Openstack-ympäristön virtuaalikoneille. Cinder-palvelun hallinnoimia tallennustiloja kutsutaan Cinder-taltioiksi. Cinder-taltiot ovat irrotettava tallennustiloja, jotka voidaan liittää Openstack-ympäristön virtuaalikoneisiin. Osa varastointipalvelun palveluista asennetaan Controller-noodille, mutta lisäksi palvelulle konfiguroidaan oma varastonoodinsa Cinder-taltioiden säilytystä varten. (Openstack Foundation 2018s, viitattu 13.4.2018.)

Cinder-palvelun asentaminen aloitetaan muiden Openstack-ympäristön palveluiden tavoin tietokannan, käyttäjän, palvelukokonaisuuden sekä loppupisteiden luomisella. Seuraavaksi Controller-noodille asennetaan Cinder-palvelun tarvitsemat paketit komennolla "apt install cinder-api cinder scheduler". Pakettien asennuksen jälkeen Cinder-palvelun konfigurointitiedostoa */etc/cinder/cinder.conf* muokataan (liite 10). Tiedostoon määritellään muiden palveluiden tavoin palvelun käyttämä tietokanta sekä yhteys identiteettipalveluun. Muokkauksen jälkeen tietokanta otetaan käyttöön komennolla "su -s /bin/sh -c "cinder-manage db sync" cinder". Lisäksi Nova konfiguroidaan käyttämään Cinder-palvelua (liite 10). Controller-noodin asennuksen viimeistelyksi tarvittavat palvelut käynnistetään uudelleen komennoilla "service nova-api restart" ja "service cinder-scheduler restart" (Openstack Foundation 2018t, viitattu 14.4.2018.)

Controller-noodin asennuksen jälkeen Block-noodilla luodaan fyysinen LVM-taltio komennolla "pvcreate /dev/sdb". Lisäksi luodaan taltior ryhmä komennolla "vgcreate cinder-volumes /dev/sdb". Seuraavaksi Block-noodille asennetaan tarvittava paketti komennolla "apt install cinder-volume". Asennuksen jälkeen Block-noodilla sijaitsevaa Cinder-palvelun konfigurointitiedostoa muokataan (liite 11). Tiedostossa määritetään muiden palveluiden tavoin muun muassa käytettävä tietokanta sekä yhteys identiteettipalveluun. Lisäksi LVM (Logical Volume Manager) kohdan alle määritetään

taltioryhmäksi luotu cinder-volumes-ryhmä. Lopuksi Cinder-palvelun palvelut käynnistetään uudelleen komennoilla "service tgt restart" ja "service cinder-volume restart". Varastointipalvelun toimintaa testataan suorittamalla komento "openstack volume service list" Controller-noodilla (kuvio 12). (Openstack Foundation 2018u, viitattu 14.4.2018.)

```
root@controller:/home/sami# openstack volume service list
```

Binary	Host	Zone	Status	State	Updated At
cinder-scheduler	controller	nova	enabled	up	2018-04-13T17:23:14.000000
cinder-volume	block@lvm	nova	enabled	up	2018-04-13T17:23:12.000000

Kuvio 12. Cinder-palvelun toiminnan testaus

4.2.6 Designate

Designate on testiympäristön DNS-palvelu (Domain Name System). Palvelun avulla verkkotunnukset, kuten `www.example.com` voidaan ohjata tiettyihin IP-osoitteisiin. Designate-palvelu käyttää pohjanaan avoimeen lähdekoodin perustuvaa DNS-palvelua BIND. Designate voidaan konfiguroida luomaan DNS-tietueita automaattisesti Nova- tai Neutron-palvelun toimintojen perusteella. (Openstack Foundation 2018y, viitattu 1.5.2018.)

Designate-palvelun asentaminen aloitetaan muiden palveluiden tavoin tietokannan, käyttäjän, palvelukokonaisuuden sekä loppupisteen luonnilla. Designate-palvelun tarvitsemat paketit asennetaan komennolla "apt-get install designate bind9 bind9utils bind9-doc". Komennon suorittamisen jälkeen asennetaan vielä Designate-palvelun osat komennolla "apt install designate-worker designate-producer designate-mdns".

Seuraavaksi luodaan RNDC-avain (Remote Name Daemon Control) sijaintiin `/etc/designate/rndc.key` komennolla "rndc-confgen -a -k designate -c /etc/designate/rndc.key". RNDC-toiminnon käyttö mahdollistaa komentokehotepohjaisen BIND-palvelun hallinnan. Avaimen luonnin jälkeen BIND-palvelun konfigurointitiedostoa `/etc/bind/named.conf.options` muokataan (liite 12). Konfigurointitiedostossa määritetään luodun RNDC-avaimen käyttö, BIND-palvelun kuuntelu osoitteet sekä DNS-tiedustelu sallitaan verkosta 10.0.0.0/24. Muokkauksen jälkeen BIND käynnistetään uudelleen komennolla "service bind9 restart".

Seuraavaksi muokataan Designate-palvelun konfigurointitiedostoa `/etc/designate/designate.conf`. Tiedostossa määritetään palvelun käyttämä tietokanta, yhteys identiteettipalveluun sekä tarvittavat laajennukset otetaan käyttöön (liite 13). Muokkausten jälkeen Designate-palvelun tietokanta otetaan käyttöön komennolla `"su -s /bin/sh -c "designate-manage database sync" designate"`. Tietokannan käyttöönoton jälkeen tarvittavat palvelut käynnistetään uudelleen komennoilla `"service designate-central restart"` ja `"service designate-api restart"`.

Designate-palvelun asennuksen viimeistelyksi luodaan `pools.yaml`-tiedosto sijaintiin `/etc/designate/pools.yaml` (liite 14). Tiedoston luonnin jälkeen suoritetaan komento `"su -s /bin/sh -c "designate-manage pool update" designate"`. Lopuksi Designate-palvelun worker-, producer- sekä mdns-palvelut käynnistetään uudelleen. (Openstack Foundation 2018z, viitattu 1.5.2018.)

Designate-palvelun toimintaa testataan uuden DNS-alueen luonnilla. DNS-alueen luonti tapahtuu komennolla `"openstack zone create --email dnspalvelu@example.com example.com."` Palvelu palauttaa vastaan ikkunan komennon suorittamisen jälkeen (kuvio 13).

Field	Value
action	CREATE
attributes	
created_at	2018-05-01T08:32:17.000000
description	None
email	dnspalvelu@example.com
id	80a90757-46e8-4249-b236-19c62905fdc9
masters	
name	example.com.
...	...

Kuvio 13. Designate-palvelun testaus

4.2.7 Horizon

Viimeinen asennettava kokonaisuus testiympäristössä on ympäristön graafinen käyttöliittymä Horizon. Graafisen käyttöliittymän käyttö ei ole ympäristössä pakollista, koska Openstack-alustan käyttö onnistuu pelkän komentokehotteen avulla. Alkuaan Horizon suunniteltiin Openstack-alustan compute-palvelun hallintaan, mutta se kehittyi lopulta nykyiseen mittaansa tukemaan useita Openstack-alustan eri palveluita. (Openstack Foundation 2018v, viitattu 14.4.2018.)

Horizon asennetaan Controller-noodille komennolla "apt install openstack-dashboard". Asennuksen jälkeen sen konfigurointitiedostoa *etc/openstack-dashboard/local_settings.py* muokataan (liite 15). Tiedossa määritellään muun muassa Horizon käyttämään Controller-noodille asennettuja palveluita sekä sen käyttö sallitaan jokaiselta isännältä. Muokkauksen jälkeen asennuksen viimeiseksi Apache-palvelimen konfigurointi ladataan uudelleen komennolla "service apache2 reload".

Asennuksen jälkeen graafiseen käyttöliittymään pääse kiinni verkkoselaimella osoitteesta <http://10.0.0.11/horizon>. Toiminnan testaamiseksi käyttöliittymään kirjaudutaan jo ennestään luodulla admin-tunnuksella (liite 16). (Openstack Foundation 2018v, viitattu 14.4.2018w.)

4.3 Ympäristön testaus

Openstack-ympäristön testaamiseksi testiympäristöön luodaan uusi projekti sekä käyttäjä nimeltä Demo, jonka alle virtuaalikoneet luodaan. Käyttäjälle luodaan admin-tunnuksen tavoin erillinen tiedosto kirjautumistietojen lataamiseksi. Seuraavaksi ympäristöön luodaan Neutron-palvelun verkkovalintoehdoksi valittu toimittajaverkko. Toimittajaverkon luonti onnistuu ainoastaan admin-tunnuksen tiedoilla, minkä vuoksi kirjautumistiedot ladataan komennolla "admin-openrc". Toimittajaverkon luonti onnistuu myös graafisen käyttöliittymän kautta, mutta tässä tapauksessa luonti on tapahtunut komennon avulla. Toimittajaverkon luonnin jälkeen verkolle luodaan aliverkko samalle osoitealueelle ympäristön noodien kanssa (liite 17).

Openstack käyttää termiä aromi virtuaalikoneiden muistin, tallennuskapasiteetin sekä resurssien tulkitsemiseen. Testauksen vuoksi Openstack-ympäristöön luodaan uusi aromi graafisen käyttöliittymän kautta (liite 18). Lisäksi käyttäjän Demo kirjautumistiedoilla luodaan SSH-avainpari julkisen avaintodennuksen käyttämiseksi (liite 17). Avainparin avulla SSH-yhteyden ottaminen Openstack-ympäristössä pystytettyihin virtuaalikoneisiin onnistuu ilman erillistä käyttäjätunnusta ja salasanaa isäntätietokoneilla, joihin kyseinen avain on tallennettu.

Lopuksi on vuorossa itse virtuaalikoneen luonti. Luonti tapahtuu Demo käyttäjän kirjautumistiedoilla. Ennen virtuaalikoneen luontia varastointipalvelun avulla luodaan yhden gigatavun taltio komennolla "openstack volume create --size 1 volume1". Itse virtuaalikoneen luonti tapahtuu komentokehoteen kautta seuraavasti (kuvio 14). Komento määrittää käytettäväksi edellä luodun aromin,

levy kuvapalveluun ladatun levykuvan, toimittajaverkon, turvallisuusryhmän, avainparin sekä virtuaalikoneen nimen. Vastaava toimenpide olisi onnistunut myös graafisen käyttöliittymän kautta (liite 19). Virtuaalikoneen luonnissa kestää hetken aikaa, mutta pian graafisesti käyttöliittymästä käsin voi varmistaa tilan nousseen aktiiviseksi. Virtuaalikone on saanut IP-osoitteen 10.0.0.50. Virtuaalikoneen luonnin jälkeen luotu taltio liitetään virtuaalikoneeseen komennolla "openstack server add volume virtuaalikone1 volume1".

```
root@controller:/home/sami# openstack server create --flavor ml.nano \
--image cirros \
--nic net-id=28fe8385-b7a4-4436-b41d-7a6cb6fc2f61 \
--security-group default --key-name mykey virtuaalikone1
```

Kuvio 14. Virtuaalikoneen luonti komentokehotteella

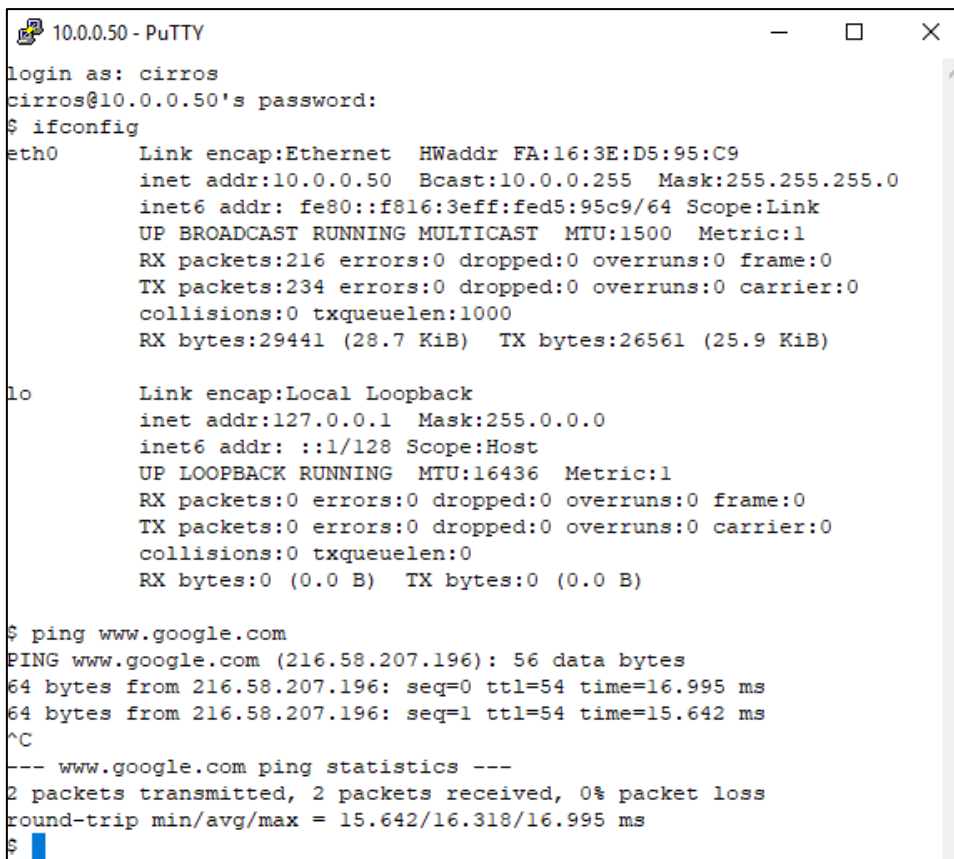
Lopulta luotua virtuaalikonetta voi tarkastella graafisen käyttöliittymän kautta (kuvio 15). Lisäksi virtuaalikoneeseen voi ottaa SSH-yhteyden (kuvio 16).

```
Connected (unencrypted) to: QEMU (instance-00000002)
[ 4.798817] cpuidle: using governor menu
[ 4.805680] EFI Variables Facility v0.08 2004-May-17
[ 4.814524] TCP cubic registered
[ 4.822930] NET: Registered protocol family 10
[ 4.847889] NET: Registered protocol family 17
[ 4.854112] Registering the dns_resolver key type
[ 4.864930] registered taskstats version 1
[ 4.921064] usb 1-1: new full-speed USB device number 2 using uhci_hcd
[ 5.249919] Magic number: 14:519:325
[ 5.257320] rtc_cmos 00:01: setting system clock to 2018-04-16 10:18:19 UTC (
1523873899)
[ 5.265695] BIOS EDD facility v0.16 2004-Jun-25, 0 devices found
[ 5.269846] EDD information not available.
[ 5.293884] Freeing unused kernel memory: 928k freed
[ 5.300627] Write protecting the kernel read-only data: 12288k
[ 5.335614] Freeing unused kernel memory: 1596k freed
[ 5.363650] Freeing unused kernel memory: 1184k freed

further output written to /dev/ttyS0

login as 'cirros' user. default password: 'cubswin:'). use 'sudo' for root.
virtuaalikone1 login: cirros
Password:
$
```

Kuvio 15. Graafisen käyttöliittymän virtuaalikoneen konsoli



```
10.0.0.50 - PuTTY
login as: cirros
cirros@10.0.0.50's password:
$ ifconfig
eth0      Link encap:Ethernet  HWaddr FA:16:3E:D5:95:C9
          inet addr:10.0.0.50  Bcast:10.0.0.255  Mask:255.255.255.0
          inet6 addr: fe80::f816:3eff:fed5:95c9/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:216 errors:0 dropped:0 overruns:0 frame:0
          TX packets:234 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:29441 (28.7 KiB)  TX bytes:26561 (25.9 KiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

$ ping www.google.com
PING www.google.com (216.58.207.196): 56 data bytes
64 bytes from 216.58.207.196: seq=0 ttl=54 time=16.995 ms
64 bytes from 216.58.207.196: seq=1 ttl=54 time=15.642 ms
^C
--- www.google.com ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 15.642/16.318/16.995 ms
$
```

Kuvio 16. SSH-yhteys PuTTYn avulla

Openstack-alustan DNS-palvelun testaamista varten testiympäristöön ladataan Ubuntu Server -levykuva, sekä luodaan uusi aromi graafisen käyttöliittymän avulla (liite 20). Ubuntu Server -levykuva soveltuu loistavasti DNS-palvelun testaamista varten, sillä levykuva sisältää jo valmiiksi tarpeelliset paketit verkkosivujen ylläpitämistä varten. Käyttöjärjestelmän asennusvaiheessa voi valita vaihtoehtoisena ohjelmistona LAMP-kokonaisuuden asentamisen. Kokonaisuus sisältää Apache http -palvelimen, jonka ylläpitämään verkkosivua DNS-palvelun testaamisessa tullaan käyttämään. Ladattuun levykuvaan perustuva virtuaalikone luodaan graafisen käyttöliittymän avulla. Virtuaalikoneen käyttöjärjestelmän asennuksen jälkeen Apache http -palvelimen oletus verkkosivun konfigurointitiedostoa `/var/www/html/index.html` muokataan seuraavanlaiseksi (liite 21). Muokkauksen jälkeen Apache-palvelin käynnistetään uudelleen komennolla `service apache2 restart`.

Seuraavaksi Demo käyttäjän kirjautumistiedoilla luodaan DNS-tietue DNS-alueen `example.com`. alle Openstack-ympäristössä. Tietueen luonti tapahtuu komennolla `openstack recordset create --records '10.0.0.51' --type A example.com. www`. Luotu tietue ohjaa verkkotunnuksen `www.example.com` IP-osoitteeseen `10.0.0.51`, joka on tässä tapauksessa ympäristöön luodun

Ubuntu Server -virtuaalikoneen IP-osoite, jossa Apache http -palvelimen ylläpitämä verkkosivu sijaitsee.

DNS-palvelun toimintaa testataan ensiksi dig-komennon avulla (liite 21). Palvelun toimintaa testataan lisäksi määrittämällä erillisen isäntätietokoneen DNS-osoitteeksi 10.0.0.11 ja menemällä verkkoselaimella osoitteeseen www.example.com palvelun toiminnan varmistamiseksi (kuvio 17).



Kuvio 17. DNS-palvelun testaus

4.4 Ympäristön hallinta

Openstack-ympäristön hallitseminen onnistuu joko graafisen käyttöliittymän tai komentokehotteen avulla. Yrityskäytössä ympäristöön lisättäisiin todennäköisesti vielä yksi noodi, jonka tehtävänä olisi keskittyä ympäristön hallintaan. Tällaisen noodin konfigurointi vaatii ainoastaan Openstack-alustan asiakasohjelman asennuksen sekä vastaavan admin-openrc-tiedoston määrittelyn (liite 1). Graafisen käyttöliittymän käyttö on taas mahdollisesti sallittu kaikilta yrityksen työasemilta. Käyttöä on tietenkin mahdollista rajata graafisen käyttöliittymän konfigurointia muuttamalla.

Käytännössä Openstack-ympäristöä voi hallita lähes täysin graafisen käyttöliittymän kautta lukuun ottamatta muutamia poikkeuksia. Uusien palveluiden vaatimien loppupisteiden luonti ei muun muassa onnistu graafisen käyttöliittymän avulla. Openstack-alustan graafinen käyttöliittymä perustuu avoimeen lähdekoodiin, minkä vuoksi tiettyjä Openstack-ympäristön palveluja ei ole edes mahdollisesti integroitu tukemaan graafista käyttöliittymää. Valmista Openstack-ympäristöä voi hallita siis graafisen käyttöliittymän tai komentokehotteen avulla. Parhaimman lopputuloksen kuitenkin saavuttaa käyttämällä molempia.

5 POHDINTA

Openstack-alustan testiympäristön asennus oli lopulta yllättävän työläs prosessi, vaikka apuna käytetty Openstack-säätöön dokumentointi on todella kattava. Ongelmia oli alussa muun muassa virtualisoitujen isäntäkoneiden verkkoyhteyden kanssa. Kyseiseltä ongelmalta olisi todennäköisesti vältetty asentamalla Openstack-alusta fyysisille isäntäkoneille. Itse Openstack-alustan konfiguroinnin kanssa ei lopulta suurempia ongelmia ollut muutamia poikkeuksia lukuun ottamatta. Kohdatut ongelmat johtuivat yleisimmin kirjoitusvirheistä palveluiden konfigurointitiedostoissa. Ongelma voi kuulostaa yksinkertaiselta, mutta sen paikantaminen voi olla työläämpää. Virheet paikannettiin pääosin Openstack-alustan lokitiedostoja selaamalla. Lokitiedostojen selailu ei kuitenkaan aina auttanut virheiden paikantamisessa, ja lopulta muun muassa verkkopalvelun asennus suoritettiin osaksi kokonaan uudelleen.

Valmis lopputulos on kuitenkin tehdyn työn arvoinen. Openstack-alustan käyttö on lopulta yllättävän yksinkertaista komentokehoteen avulla, ja hallintaan käytettävät komennot muodostuvat yleensä aina tiettyä rakennetta seuraten. Graafinen käyttöliittymä antaa taas ympäristön käyttäjällä tutun ja turvallisen kuvan ympäristön hallinnasta. Esimerkiksi virtuaalikoneiden luonti onnistuu graafisen käyttöliittymän avulla muutaman klikkauksen avulla.

Openstack-alustan tulevaisuus voidaan nähdä vakaana sen suuren kehittäjäyhteisön sekä alustan merkittävien rahoittajien vuoksi (Openstack Foundation 2018a, viitattu 2.5.2018). Lukuisat yritykset todennäköisesti pohtivat oman infrastruktuurinsa päivittämistä. Openstack-alustan integrointi yrityksen omaan ympäristöön on mahdollisesti yksi näistä lukuisista eri vaihtoehdoista infrastruktuurin päivittämisen osalta.

Tehty Openstack-alustan testiympäristön asennus mahdollisti perehtymisen Openstack-alustan toimintaan ja uusien taitojen oppimisen. Opittuja taitoja jotka koskevat Openstack-ympäristön ylläpitoa tai konfigurointia voidaan todennäköisesti käyttää hyödyksi työelämässä. Seuraava etappi Openstack-ympäristön toimintaan tutustuessa olisi ympäristön konfigurointi fyysisillä isäntäkoneilla, sekä ympäristön integroimien julkisen pilven kanssa Omni-projektin avulla. Omni-projektin avulla Openstack-alusta muuntuu niin sanotuksi hybridipilveksi (Kapoor 2018, viitattu 2.5.2018).

LÄHTEET

Chrony 2017. Introduction. Viitattu 26.4.2018, <https://chrony.tuxfamily.org/>.

Chrzanowska, N 2016. 10 Reasons why SaaS model is beneficial for vendors and clients. Viitattu 28.3.2018, <https://brand24.com/blog/10-reasons-why-saas-model-is-beneficial-for-vendors-and-clients/>.

Citrix 2014. What is hybrid cloud? Viitattu 25.4.2018, <https://www.citrix.fi/glossary/hybrid-cloud.html>.

Dolph, M 2015. Openstack keystone fernet tokens. Viitattu 3.4.2018, <http://blog.dolphm.com/openstack-keystone-fernet-tokens/>.

Dignan, L 2018. Top cloud providers 2018. Viitattu 16.5.2018, <https://www.zdnet.com/article/cloud-providers-ranking-2018-how-aws-microsoft-google-cloud-platform-ibm-cloud-oracle-alibaba-stack/>.

Dusablon, É 2017. Cloud - NDT - Presentation, Dia 8/25. Viitattu 26.4.2018, <https://www.slideshare.net/ricDusablon/cloud-ndt-presentation>.

Heino, P. 2010. Pilvipalvelut. Hämeenlinna: Talentum Media Oy.

Interoute 2018. What is private cloud? Viitattu 26.3.2018, <https://www.interoute.com/what-private-cloud>.

Joshi, S. 2014. What is platform as a service (PaaS)? Viitattu 28.3.2018 <https://www.ibm.com/blogs/cloud-computing/2014/02/what-is-platform-as-a-service-paas/>.

Kapoor, A. 2018. Openstack Omni, the Open-Source Alternative to VMware and AWS for Hybrid Cloud. Viitattu 2.5.2018, <https://platform9.com/blog/openstack-omni-the-open-source-alternative-to-vmware-aws-for-hybrid-cloud/>.

Microsoft 2018a. What is cloud computing? Viitattu 20.3.2018,
<https://azure.microsoft.com/en-in/overview/what-is-cloud-computing/>.

Microsoft 2018b. Azure pricing. Viitattu 28.4.2018, <https://azure.microsoft.com/en-gb/pricing/>.

Microsoft 2018c. What is a private cloud? Viitattu 25.4.2018,
<https://azure.microsoft.com/en-us/overview/what-is-a-private-cloud/>.

Microsoft 2018d. What is Azure Stack? Viitattu 16.5.2018
<https://azure.microsoft.com/en-us/overview/azure-stack/>.

Opensource 2018. What is Openstack. Viitattu 29.3.2018,
<https://opensource.com/resources/what-is-openstack/>.

Openstack Foundation 2018a. Open source software for creating private and public clouds. Viitattu 29.3.2018, <https://www.openstack.org>.

Openstack Foundation 2018b. Introduction: A Bit of OpenStack History. Viitattu 29.3.2018,
<https://docs.openstack.org/project-team-guide/introduction.html>.

Openstack Foundation 2018c. Openstack releases. Viitattu 29.3.2018,
<https://releases.openstack.org/>.

Openstack Foundation 2018d. Install Openstack services. Viitattu 2.4.2018,
<https://docs.openstack.org/install-guide/openstack-services.html>.

Openstack Foundation 2018e. Identity Service Overview. Viitattu 3.4.2018,
<https://docs.openstack.org/keystone/pike/install/get-started-ubuntu.html>.

Openstack Foundation 2018f. Keystone Architecture. Viitattu 3.4.2018,
<https://docs.openstack.org/keystone/queens/getting-started/architecture.html>.

Openstack Foundation 2018g. Image Service Overview. Viitattu 4.4.2018,
<https://docs.openstack.org/glance/pike/install/get-started.html>.

Openstack Foundation 2018h. Nova System Architecture. Viitattu 11.4.2018,
<https://docs.openstack.org/nova/latest/user/architecture.html>.

Openstack Foundation 2018i. Placement API. Viitattu 11.4.2018,
<https://docs.openstack.org/nova/latest/user/placement.html>.

Openstack Foundation 2018j. Install and configure. Viitattu 6.4.2018,
<https://docs.openstack.org/keystone/pike/install/keystone-install-ubuntu.html#install-and-configure-components>.

Openstack Foundation 2018k. Install and configure (Ubuntu). Viitattu 7.4.2018,
<https://docs.openstack.org/glance/pike/install/install-ubuntu.html>.

Openstack Foundation 2018l. Install and configure controller node for Ubuntu. Viitattu 11.4.2018,
<https://docs.openstack.org/nova/pike/install/controller-install-ubuntu.html>.

Openstack Foundation 2018m. Install and configure a compute node for Ubuntu. Viitattu 12.4.2018,
<https://docs.openstack.org/nova/pike/install/compute-install-ubuntu.html>.

Openstack Foundation 2018n. Networking service overview. Viitattu 12.4.2018,
<https://docs.openstack.org/neutron/latest/install/common/get-started-networking.html>.

Openstack Foundation 2018o. Overview. Viitattu 12.4.2018,
<https://docs.openstack.org/neutron/latest/install/overview.html>.

Openstack Foundation 2018p. Metadata Service Support. Viitattu 12.4.2018,
https://docs.openstack.org/dragonflow/latest/specs/metadata_service.html.

Openstack Foundation 2018q. Install and configure controller node. Viitattu 12.4.2018,
<https://docs.openstack.org/neutron/pike/install/controller-install-ubuntu.html>.

Openstack Foundation 2018r. Install and configure compute node. Viitattu 12.4.2018,
<https://docs.openstack.org/neutron/pike/install/compute-install-ubuntu.html>.

Openstack Foundation 2018s. Cinder Installation Tutorial. Viitattu 13.4.2018,
<https://docs.openstack.org/cinder/pike/install/>.

Openstack Foundation 2018t. Install and configure controller node. Viitattu 14.4.2018,
<https://docs.openstack.org/cinder/pike/install/cinder-controller-install-ubuntu.html>.

Openstack Foundation 2018u. Install and configure a storage node. Viitattu 14.4.2018,
<https://docs.openstack.org/cinder/pike/install/cinder-storage-install-ubuntu.html>.

Openstack Foundation 2018v. Horizon Basics. Viitattu 14.4.2018,
<https://docs.openstack.org/horizon/latest/contributor/intro.html#contributor-intro>.

Openstack Foundation 2018w. Install and configure for Ubuntu. Viitattu 14.4.2018,
<https://docs.openstack.org/horizon/pike/install/install-ubuntu.html>.

Openstack Foundation 2018x. Environment. Viitattu 24.4.2018,
<https://docs.openstack.org/install-guide/environment.html>.

Openstack Foundation 2018y. Designate, a DNSaaS component for Openstack. Viitattu 1.5.2018,
<https://docs.openstack.org/designate/latest/>.

Openstack Foundation 2018z. Install and configure for Ubuntu. Viitattu 1.5.2018,
<https://docs.openstack.org/designate/pike/install/install-ubuntu.html>.

Openstack Foundation 2018å. Companies Supporting The OpenStack Foundation. Viitattu 2.5.2018, <https://www.openstack.org/foundation/companies/>.

Openstack Foundation 2018ä. Release Management. Viitattu 4.5.2018,
<https://docs.openstack.org/project-team-guide/release-management.html>.

Openstack Foundation 2018ö. Conceptual architecture. Viitattu 4.5.2018,
<https://docs.openstack.org/install-guide/get-started-conceptual-architecture.html>.

Rackspace 2018. Openstack – from cloud pioneer to industry innovator. Viitattu 26.3.2018, <https://www.rackspace.com/openstack>.

Salesforce 2018. Sell, service, and market smarter with the world's #1 CRM platform. Viitattu 26.4.2018, <https://www.salesforce.com/fi/>.

Salo, I. 2012. Hyötyä pilvipalveluista. Jyväskylä: Docento.

Shameen 2016. Openstack Glance. Viitattu 4.4.2018, <https://www.sparkmycloud.com/blog/openstack-glance/>.

Snedeker, B. 2017. Cloud Computing Pros and Cons for Small Business. Viitattu 24.4.2018, <https://www.infusionsoft.com/business-success-blog/growth/planning-strategy/the-pros-and-cons-of-public-and-private-clouds>.

StateTech 2017. 5 Important benefits of Infrastructure as a Service. Viitattu 28.3.2018, <https://statetechmagazine.com/article/2014/03/5-important-benefits-infrastructure-service>.

Suse 2016. New Research Shows OpenStack Adoption Strong, But Complexities Remain. Viitattu 17.5.2018, <https://www.suse.com/c/news/new-research-shows-openstack-adoption-strong-but-complexities-remain/>.

TechTarget 2015. Pay-as-you-go cloud computing (PAYG cloud computing). Viitattu 20.3.2018, <http://searchstorage.techtarget.com/definition/pay-as-you-go-cloud-computing-PAYG-cloud-computing>.

Techtarget 2017. Private cloud (internal cloud or corporate cloud). Viitattu 26.3.2018, <http://searchcloudcomputing.techtarget.com/definition/private-cloud>.

LIITELUETTELO

	sivu
Liite 1 Identiteettipalvelun konfigurointi	39
Liite 2 Levykuvapalvelun konfigurointi	40
Liite 3 Levykuvan lataus	41
Liite 4 Nova-palvelun konfigurointi 1	42
Liite 5 Nova-palvelun konfigurointi 2	43
Liite 6 Verkkopalvelun konfigurointi 1	45
Liite 7 Liitännäisten konfigurointi	46
Liite 8 Verkkopalvelun konfigurointi 2	47
Liite 9 Verkkopalvelun konfigurointi 3	48
Liite 10 Varastointipalvelun konfigurointi 1	49
Liite 11 Varastointipalvelun konfigurointi 2	50
Liite 12 BIND-palvelun konfigurointi	51
Liite 13 Designate-palvelun konfigurointi	52
Liite 14 Pools.yaml-tiedosto	53
Liite 15 Graafisen käyttöliittymään konfigurointi	54
Liite 16 Horizon-kirjautumisikkuna	55
Liite 17 Toimittajaverkon ja SSH-avainparin luonti	56
Liite 18 Aromin luonti	57
Liite 19 Virtuaalikoneen luonti	58
Liite 20 Aromin luonti ja levykuvan lataus	59
Liite 21 DNS-palvelun testaus	60

Lähes jokainen Openstack-alustan palvelua vaati tietokannan tietojen säilyttämiseen. Luodulle tietokannalle annetaan tarvittavat oikeudet Keystone nimiselle käyttäjälle. Identiteettipalvelu myös otetaan käyttöön ja admin-tunnukselle määritetään salasana.

Keystone-tietokannan luonti:

```
root@controller:/home/sami# mysql
MariaDB [(none)]> CREATE DATABASE keystone;

MariaDB [(none)]> GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'localhost'
IDENTIFIED BY 'keystone';

MariaDB [(none)]> GRANT ALL PRIVILEGES ON keystone.* TO 'keystone'@'%'
IDENTIFIED BY 'keystone';
```

Fernet-avain arkistojen alustus:

```
root@controller:/home/sami # keystone-manage fernet_setup --keystone
-user keystone --keystone-group keystone

root@controller:/home/sami # keystone-manage credential_setup
--keystone-user keystone --keystone-group keystone
```

Identiteettipalvelun käyttöönotto:

```
root@controller:/home/sami# keystone-manage bootstrap \
--bootstrap-password salasana \
--bootstrap-admin-url http://controller:35357/v3/ \
--bootstrap-internal-url http://controller:5000/v3/ \
--bootstrap-public-url http://controller:5000/v3/ \
--bootstrap-region-id RegionOne
```

Admin-openrc -tiedoston sisältö:

```
export OS_PROJECT_DOMAIN_NAME=Default
export OS_USER_DOMAIN_NAME=Default
export OS_PROJECT_NAME=admin
export OS_USERNAME=admin
export OS_PASSWORD=salasana
export OS_AUTH_URL=http://controller:35357/v3
export OS_IDENTITY_API_VERSION=3
export OS_IMAGE_API_VERSION=2
```

Loppupisteet ovat osoitteita joiden avulla Openstack-alustan palvelut keskustelevat toistensa kanssa. Loppupisteet luodaan yleisimmin public-, internal- sekä admin-liitännöille. Levykuvapalvelun konfigurointitiedossa määritetään Glance käyttämään sille luotua tietotankaa sekä yhteys identiteettipalveluun

Glance-palvelun loppupisteiden luonti:

```
root@controller:/home/sami # openstack endpoint create --region RegionOne image
public http://controller:9292

root@controller:/home/sami # openstack endpoint create --region RegionOne image
internal http://controller:9292

root@controller:/home/sami # openstack endpoint create --region RegionOne image
admin http://controller:9292
```

Glance-palvelun konfigurointitiedoston muokkaus:

```
[database]
...
connection = mysql+pymysql://glance:glance@controller/glance

[keystone_authtoken]
...
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = glance
password = salasana

[paste_deploy]
...
flavor = keystone

[glance_store]
...
stores = file,http
default_store = file
filesystem_store_datadir = /var/lib/glance/images/
```


Levykuvia Openstack-ympäristöön ladatessa on määriteltävä mm. minkä tyyppinen levykuva on sekä sen näkyvyys. Cirros-levykuvat ovat kooltaan pieniä ja soveltuvat hyvin alustan testaamiseen. Lisäksi komennon suorittamisen jälkeen on tarkistettu, että levykuvan lataus onnistui.

```

root@controller:/home/sami # openstack image create "cirros" \
  --file cirros-0.3.5-x86_64-disk.img \
  --disk-format qcow2 --container-format bare \
  --public

```

Field	Value
checksum	f8ab98ff5e73ebab884d80c9dc9c7290
container_format	bare
created_at	2018-04-07T08:02:08Z
disk_format	qcow2
file	/v2/images/4ead9dc6-ebd5-486f/file
id	4ead9dc6-ebd5-486f-87af-e986dc27535b
min_disk	0
min_ram	0
name	cirros
owner	1741a36d662a44a2ade71ad6e40e96c1
protected	False
schema	/v2/schemas/image
size	13267968
status	active
tags	
updated_at	2018-04-07T08:02:08Z
virtual_size	None
visibility	public

```

root@controller:/home/sami# openstack image list

```

ID	Name	Status
4ead9dc6-ebd5-486f-87af-e986dc27535b	cirros	active

Nova-palvelulle on luotu kolme tietokantaa, joille kaikille on annettu tarvittavat oikeudet Nova nimiselle käyttäjälle. Lisäksi compute- sekä placement-palveluille on luotu loppupisteet public-, internal- sekä admin-liitännöille. Lopuksi Nova-palvelun tietokannat on otettu käyttöön.

Tietokantojen luonti:

```
root@controller:/home/sami# mysql
MariaDB [(none)]> CREATE DATABASE nova_api;
MariaDB [(none)]> CREATE DATABASE nova;
MariaDB [(none)]> CREATE DATABASE nova_cell0;
```

Oikeuksien antaminen:

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON Tietokannan_nimi.* TO
'nova'@'localhost' IDENTIFIED BY 'salasana';

MariaDB [(none)]> GRANT ALL PRIVILEGES ON Tietokannan_nimi.* TO 'nova'@'%'
IDENTIFIED BY 'salasana';
```

Esimerkki Novan loppupisteen luomisesta:

```
root@controller:/home/sami# openstack endpoint create --region RegionOne compute
liitännän_tyyppi http://controller:8774/v2.1
```

Esimerkki Placement APIn loppupisteen luomisesta:

```
openstack endpoint create --region RegionOne placement liitännän_tyyppi
http://controller:8778
```

liitännän_tyyppi = public tai internal tai admin

Tietokannan käyttöönotto:

```
root@controller:/home/sami# su -s /bin/sh -c "nova-manage api_db sync" nova
```

Cell0-tietokannan rekisteröinti:

```
root@controller:/home/sami # su -s /bin/sh -c "nova-manage cell_v2 map_cell0"
nova
```

Cell1-solun luonti:

```
root@controller:/home/sami # su -s /bin/sh -c "nova-manage cell_v2 create_cell -
-name=cell1 --verbose" nova
```

Nova-palvelun konfigurointitiedossa Controller-noodilla määritetään sen käyttämät tietokannat, IP-osoite, Neutron-palvelun käyttö, VNC-protokollan käyttö, Glance- sekä placement-palvelun käyttö. Lisäksi Compute-noodilla on määritetty käytettäväksi virtualisoinnin tyypiksi QEMU.

```
[api_database]
# ...
connection = mysql+pymysql://nova:salasana@controller/nova_api

[database]
# ...
connection = mysql+pymysql://nova:salasana@controller/nova

[DEFAULT]
# ...
transport_url = rabbit://openstack:salasana@controller

# ...
my_ip = 10.0.0.11

# ...
use_neutron = True
firewall_driver = nova.virt.firewall.NoopFirewallDriver

[api]
# ...
auth_strategy = keystone

[keystone_authtoken]
# ...
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = nova
password = salasana

[vnc]
enabled = true
# ...
vncserver_listen = $my_ip
vncserver_proxyclient_address = $my_ip
```

```
[glance]
# ...
api_servers = http://controller:9292

[oslo_concurrency]
# ...
lock_path = /var/lib/nova/tmp

[placement]
# ...
os_region_name = RegionOne
project_domain_name = Default
project_name = service
auth_type = password
user_domain_name = Default
auth_url = http://controller:35357/v3
username = placement
password = salasana
```

Compute noodilla tehtävät lisä muutokset:

```
[DEFAULT]
# ...
my_ip = 10.0.0.22

[vnc]
# ...
enabled = True
vncserver_listen = 0.0.0.0
vncserver_proxyclient_address = $my_ip
novncproxy_base_url = http://controller:6080/vnc_auto.html
```

Virtualisoinnin tyyppin vaihto:

```
[libvirt]
# ...
virt_type = qemu
```

Neutron-palvelun konfigurointi ei juurikaan eroa periaatteeltaan Glance-palvelun tai Nova-palvelun kanssa. Konfigurointitiedostossa määritetään käytettävä tietokanta, ML2-liitännäisen käyttöönotto sekä yhteys identiteettipalveluun.

```
[database]
# ...
connection = mysql+pymysql://neutron:salasana@controller/neutron

[DEFAULT]
# ...
core_plugin = ml2
service_plugins =

# ...
transport_url = rabbit://openstack:salasana@controller

# ...
auth_strategy = keystone

# ...
notify_nova_on_port_status_changes = true
notify_nova_on_port_data_changes = true

[keystone_authtoken]
# ...
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = neutron
password = salasana
```

Neutron vaatii useiden liitännäisten konfiguroinnin toimiakseen. Modular Layer 2 -liitännäinen mahdollistaa virtuaalisten kytkimien käytön ympäristössä. Linux Bridge Agent -liitännäistä käytetään taas virtuaalisten verkkoinfrastruktuurien rakentamiseen. Lisäksi DHCP-palvelu otetaan käyttöön ympäristössä ajettaville virtuaalikoneille.

Modular Layer 2 -liitännäinen /etc/neutron/plugins/ml2/ml2_conf.ini

```
[ml2]
# ...
type_drivers = flat,vlan

# ...
tenant_network_types =

# ...
mechanism_drivers = linuxbridge

# ...
extension_drivers = port_security

[ml2_type_flat]
# ...
flat_networks = provider

[securitygroup]
# ...
enable_ipset = true
```

Linux Bridge Agent -liitännäinen /etc/neutron/plugins/ml2/linuxbridge_agent.ini

```
[linux_bridge]
physical_interface_mappings = enp0s8

[vxlan]
enable_vxlan = false

[securitygroup]
# ...
enable_security_group = true
firewall_driver = neutron.agent.linux.iptables_firewall.IptablesFirewallDriver
```

DHCP Agent /etc/neutron/dhcp_agent.ini

```
[DEFAULT]
# ...
interface_driver = linuxbridge
dhcp_driver = neutron.agent.linux.dhcp.Dnsmasq
enable_isolated_metadata = true
```

Openstack-ympäristön virtuaalikoneet käyttävät Metadata Agent -liitännäistä virtuaalikonekohtaisen tietojen noutamiseen. Lisäksi Nova konfiguroidaan käyttämään Neutron-palvelua sekä Neutron-palvelun tietokanta otetaan käyttöön.

```
Metadata Agent -liitännäinen /etc/neutron/metadata_agent.ini

# ...
nova_metadata_host = controller
metadata_proxy_shared_secret = salaisuus
```

Novan konfigurointitiedoston muokkaukset:

```
[neutron]
# ...
url = http://controller:9696
auth_url = http://controller:35357
auth_type = password
project_domain_name = default
user_domain_name = default
region_name = RegionOne
project_name = service
username = neutron
password = salasana
service_metadata_proxy = true
metadata_proxy_shared_secret = salaisuus
```

Tietokannan käyttöönotto:

```
su -s /bin/sh -c "neutron-db-manage --config-file \ /etc/neutron/neutron.conf --
config-file \ /etc/neutron/plugins/ml2/ml2_conf.ini upgrade head" neutron
```

Palvelujen uudelleen käynnistys:

```
root@controller:/home/sami# service nova-api restart
root@controller:/home/sami# service neutron-linuxbridge-agent restart
root@controller:/home/sami# service neutron-server restart
root@controller:/home/sami# service neutron-dhcp-agent restart
root@controller:/home/sami# service neutron-metadata-agent restart
```

Compute-noodilla tehtävät muutokset eivät juurikaan eroa Controller-noodille tehtyjen muutosten kanssa. Konfigurointitiedossa määritetään yhteys identiteettipalveluun sekä Nova konfiguroidaan käyttämään Neutron-palvelua

Neutronin konfigurointi:

```
[DEFAULT]
# ...
transport_url = rabbit://openstack:salasana@controller

# ...
auth_strategy = keystone

[keystone_authtoken]
# ...
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = neutron
password = salasana
```

Novan konfigurointi:

```
[neutron]
# ...
url = http://controller:9696
auth_url = http://controller:35357
auth_type = password
project_domain_name = default
user_domain_name = default
region_name = RegionOne
project_name = service
username = neutron
password = salasana
```


Cinder-palvelun konfigurointi ei periaatteeltaan eroa muista palveluista. Konfigurointitiedossa määritellään mm. käytettävä tietokanta sekä yhteys identiteettipalveluun. Lisäksi Nova konfiguroidaan käyttämään Cinder-palvelua.

```
[database]
# ...
connection = mysql+pymysql://cinder:salasana@controller/cinder

[DEFAULT]
# ...
transport_url = rabbit://openstack:salasana@controller

# ...
auth_strategy = keystone

# ...
my_ip = 10.0.0.11

[keystone_authtoken]
# ...
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = cinder
password = salasana

[oslo_concurrency]
# ...
lock_path = /var/lib/cinder/tmp
```

```
/etc/nova/nova.conf
```

```
[cinder]
# ...
os_region_name = RegionOne
```

```
[database]
# ...
connection = mysql+pymysql://cinder:salasana@controller/cinder

[DEFAULT]
# ...
transport_url = rabbit://openstack:salasana@controller

# ...
auth_strategy = keystone

# ...
my_ip = 10.0.0.33

# ...
enabled_backends = lvm

# ...
glance_api_servers = http://controller:9292

[keystone_authtoken]
# ...
auth_uri = http://controller:5000
auth_url = http://controller:35357
memcached_servers = controller:11211
auth_type = password
project_domain_name = default
user_domain_name = default
project_name = service
username = cinder
password = salasana

[lvm]
# ...
volume_driver = cinder.volume.drivers.lvm.LVMVolumeDriver
volume_group = cinder-volumes
iscsi_protocol = iscsi
iscsi_helper = tgtadm

[oslo_concurrency]
# ...
lock_path = /var/lib/cinder/tmp
```

BIND on avoimeen lähdekoodiin perustuva DNS-palvelu. Designate-palvelu käyttää BIND-palvelua pohjana DNS-alueiden sekä tietueiden hallinnassa. Designate voidaan myös konfiguroida käyttämään muita avoimen lähdekoodin DNS-palveluita BIND-palvelun lisäksi.

```
include "/etc/designate/rndc.key";

options {
    directory "/var/cache/bind";

    allow-new-zones yes;
    request-ixfr no;
    listen-on port 53 { 127.0.0.1; 10.0.0.11; };
    recursion no;
    allow-query { 127.0.0.1; 10.0.0.0/24; };

    forwarders {
        8.8.8.8;
        8.8.4.4;
    };

    dnssec-validation auto;

    auth-nxdomain no;
    listen-on-v6 { any; };
};

controls {
    inet 127.0.0.1 port 953
        allow { 127.0.0.1; } keys { "designate"; };
};
```

Designate-palvelun konfigurointitiedosto löytyy sijainnista `/etc/designate/designate.conf`. Tiedostoon tehtävät muokkaukset ovat samankaltaiset muiden Openstack-ympäristön palveluiden kanssa.

```
[service:api]
listen = 0.0.0.0:9001
auth_strategy = keystone
enable_api_v1 = True
api_base_uri = http://controller:9001/
enabled_extensions_v1 = quotas, reports
enable_api_v2 = True
enabled_extensions_v2 = quotas, reports
...

[keystone_authtoken]
auth_host = controller
auth_port = 35357
auth_protocol = http
admin_tenant_name = service
admin_user = designate
admin_password = salasana
...

[service:worker]
enabled = True
notify = True
...

[storage:sqlalchemy]
connection = mysql+pymysql://designate:salasana@controller/designate
```

```
- name: default

attributes: {}

ns_records:
- hostname: ns1-1.example.org.
  priority: 1

nameservers:
- host: 127.0.0.1
  port: 53

targets:
- type: bind9
  description: BIND9 Server 1

masters:
- host: 10.0.0.11
  port: 5354

options:
  host: 127.0.0.1
  port: 53
  rndc_host: 127.0.0.1
  rndc_port: 953
  rndc_key_file: /etc/designate/rndc.key
```

Openstack-alustan graafisen käyttöliittymän konfigurointitiedostossa sen käyttö on sallittu kaikilta isänniltä. Lisäksi käyttöliittymän kautta luotujen käyttäjien oletus rooliksi on määritetty "user".

```
# ...
OPENSTACK_HOST = "controller"

# ...
ALLOWED_HOSTS = ['*']

# ...
SESSION_ENGINE = 'django.contrib.sessions.backends.cache'

CACHES = {
    'default': {
        'BACKEND': 'django.core.cache.backends.memcached.MemcachedCache',
        'LOCATION': 'controller:11211',
    }
}

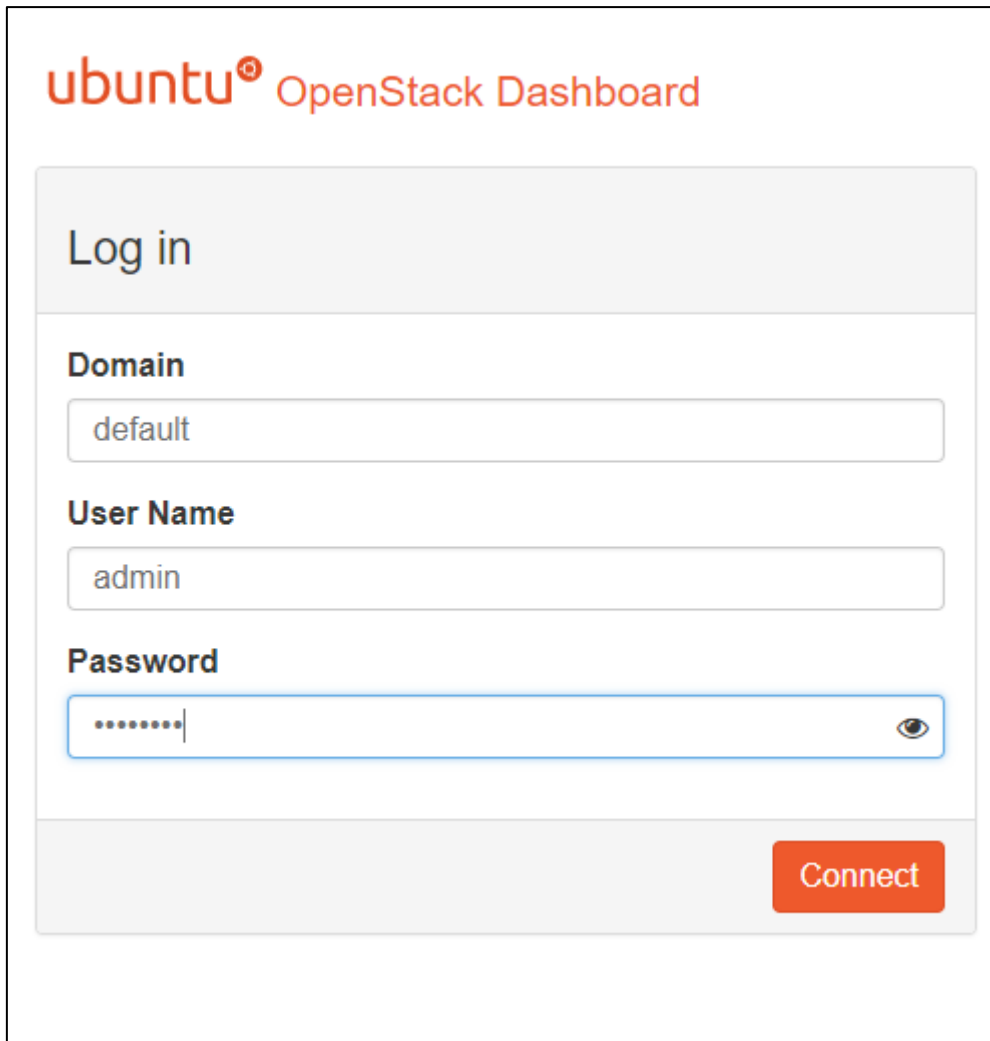
# ...
OPENSTACK_KEYSTONE_URL = "http://%s:5000/v3" % OPENSTACK_HOST

# ...
OPENSTACK_KEYSTONE_MULTIDOMAIN_SUPPORT = True

# ...
OPENSTACK_KEYSTONE_DEFAULT_DOMAIN = "Default"

# ...
OPENSTACK_KEYSTONE_DEFAULT_ROLE = "user"
```

Graafiseen käyttöliittymään pääsee kiinni verkkoselaimella osoitteesta <http://10.0.0.11/horizon>



The image shows the login interface of the Ubuntu OpenStack Dashboard. At the top, the logo "ubuntu" in orange is followed by "OpenStack Dashboard" in a lighter orange font. Below this is a light gray header bar with the text "Log in". The main form area contains three input fields: "Domain" with the value "default", "User Name" with the value "admin", and "Password" with masked characters ".....". To the right of the password field is an eye icon for toggling visibility. At the bottom right of the form is an orange "Connect" button.

ubuntu[®] OpenStack Dashboard

Log in

Domain

default

User Name

admin

Password

.....

Connect

Toimittajaverkon luonti:

```
root@controller:/home/sami/# openstack network create --share --external \  
--provider-physical-network provider \  
--provider-network-type flat provider
```

Aliverkon luonti:

```
root@controller:/home/sami/# openstack subnet create --network provider \  
--allocation-pool start=10.0.0.40,end=10.0.0.253 \  
--dns-nameserver 8.8.8.8 --gateway 10.0.0.1 \  
--subnet-range 10.0.0.0/24 provider
```

Avainparin luonti:

```
root@controller:/home/sami/# ssh-keygen -q -N ""  
root@controller:/home/sami/# openstack keypair create --public-key  
~/.ssh/id_rsa.pub mykey
```


Create Flavor

Flavor Information

Flavor Access

Name

nano

ID

auto

VCPUs

1

RAM (MB)

64

Root Disk (GB)

1

Ephemeral Disk (GB)

0

Swap Disk (MB)

0

RX/TX Factor

1

Flavors define the sizes for RAM, disk, number of cores, and other resources and can be selected when users deploy instances.

Cancel

Create Flavor

Launch Instance

Details

Source

Flavor

Networks

Network Ports

Security Groups

Key Pair

Configuration

Server Groups

Scheduler Hints

Metadata

Please provide the initial hostname for the instance, the availability zone where it will be deployed, and the instance count. Increase the Count to create multiple instances with the same settings.

Instance Name

Availability Zone

nova

Count

1

Total Instances
(10 Max)

30%

2 Current Usage

1 Added

7 Remaining

Cancel

< Back

Next >

Launch Instance

Ubuntu Server -levykuvan lataus ympäristöön:

Create Image

Image Details

Image Details

Specify an image to upload to the Image Service.

Image Name*

Ubuntu-server

Image Description

Image Source

Source Type

File

File*

Browse...

ubuntu-16.04.4-server-amr

Aromin luonti:

Create Flavor

Flavor Information

Flavor Access

Name *

m1.medium

ID ?

auto

VCPUs *

2

RAM (MB) *

1024

Root Disk (GB) *

Flavors define the sizes for RAM, disk, number of cores, and other resources and can be selected when users deploy instances.

Apache http -palvelimen ylläpitämän verkkosivun rakenne:

```
/var/www/html/index.html -tiedoston sisältö

html>

<head>
  <title>Openstack-pilvipalvelualusta</title>
</head>

<body>

<h2>Openstack-alustan DNS-palvelun testaus</h2>

</body>

</html>
```

Dig-komennon käyttö:

```
root@controller:/home/sami# dig www.example.com.

; <<>> DiG 9.10.3-P4-Ubuntu <<>> www.example.com.
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21332
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:: udp: 4096
;; QUESTION SECTION:
;www.example.com.                IN      A

;; ANSWER SECTION:
www.example.com.                 3600    IN      A      10.0.0.51

;; AUTHORITY SECTION:
example.com.                     3600    IN      NS      ns1-1.example.org.

;; Query time: 0 msec
;; SERVER: 10.0.0.11#53(10.0.0.11)
;; WHEN: Mon Apr 30 20:08:26 EEST 2018
;; MSG SIZE rcvd: 91
```